Global Information Assurance Certification Paper

Security Analysis of AFS Client Workstations

Milind Saraph
Office of Information Technologies
University of Notre Dame

August 15, 2000

Abstract:

This report analyzes the vulnerabilities of identically configured clusters of over 200 Sun workstations geographically located at four different locations on the University of Notre Dame campus. It also outlines the steps taken to address some of these and lists possible solutions for the remaining ones. This analysis is performed as part of the overall task of upgrading all the workstations from Solaris 2.6 to Solaris 7 running in 64 bit mode.

Due to identical configuration, any misconfiguration, missteps or any compromise on one workstation implies potential vulnerability of over 200 workstations. The University of Notre Dame runs a medium sized AFS (Andrew File System) cell and all the workstations are AFS clients. All the application software, important configuration files and user home directories reside in AFS. AFS uses Kerberos4 based authentication and provides per directory access control lists.

From security perspective, AFS in authors's opinion undoubtedly offers a superior solution. The section on administrative practices and the summary section have been truncated.

Contents

- Contents
- 1. Purpose
- 2. Environment
  - 2.1 Physical Environment
  - 2.2 Software Environment
  - 2.3 Hardware Configuration
- 3. AFS: Andrew File System
  - 3.1 Overview
  - 3.2 AFS File System
  - 3.3 AFS System Software
  - 3.4 AFS Package Utility
  - 3.5 Implications of root compromise on an AFS Client
- 4. Installation Procedure
  - 4.1 Install Solaris 7
  - 4.2 Install AFS
  - 4.3 Customization
  - 4.4 Create Master Disks
  - 4.5 Uprade Workstations
- 5. Security Analysis of Prototype Workstation
- 6. Security Analysis of Cluster Workstation
  - 6.1 Physical
  - 6.2 Host based
  - 6.3 Risks from Third Party Software
  - 6.4 Network Services
  - 6.5 Sendmail
  - 6.6 Administrative Practices
  - 6.7 Backup Procedures
  - 6.8 Disaster Recovery
  - 6.9 Summary
- Bibliography

1. Purpose

This report analyzes the vulnerabilities of identically configured clusters of over 200 Sun workstations geographically located at four different locations on the University of Notre Dame campus. It also outlines the steps taken to address some of these and lists possible solutions for the remaining ones. This analysis is performed as part of the overall task of upgrading all the workstations from Solaris 2.6 to Solaris 7 running in 64 bit mode.

This reports is submitted as a partial requirement for GIAC Unix Security Analyst certification. I do not wish to see this published or posted without anonymizing.

2. Environment

The following sections describe the physical environment, hardware configuration and software environment to provide a context to the analysis.

2.1 Physical Environment

The Office of Information Technologies (OIT) is responsible for acquiring, installing, configuring and administering clusters of Unix workstations which are primarily used in undergraduate/graduate education and secondorily used in graduate research. A large majority, about 90 percent of these workstations are Sun workstations with a small percentage being SGI workstations. At this time, there are 199 Sun workstations in student labs located in three different building and 10 instructor workstations in one building. An enterprise class machine provides the shell services for remote users and those without Unix machine on their desk.

2.2 Software Environment

The University of Notre Dame (ND) runs a medium size AFS [1] cell and consists of 18 file servers with over a terrabyte of mirrored data. The institutional file space is accessed by Unix workstations, PCs and Macs through different mechanisms. The Unix and WinNT machines run Transarc AFS clients whereas Win9x machines and Macs access the files space via AFS/NFS translators, AFS/ATalk translators and through Samba. All login, web and mail authentication is based on AFS Kerberos4. From this point on all the description, comments and remarks refer to Unix workstations.

All the user home directories and application software resides in AFS. The department and college staff members who need to install software for individual departments/colleges, and who are not part of the OIT, are allocated space in the AFS tree and are given rights to administer that space.

All the Unix workstations NFS mount the /var/spool/mail directory exported by a central mail server machine which provides mail service only to the Unix workstations. This is logically and physically distinct from campus mail server machine.

2.3 Hardware Configuration

Each Sun workstation is a single processor (296 MHZ), Ultra 3D workstation with 128 MB memory and 4.2 GB internal disk. The disk is partitioned as follows:

```
/                   2.0   GB       (root and usr)
swap                1.5   GB
/usr/vice/cache     0.512 GB       (AFS Cache)
```

Each workstation has a 10/100 Mbps network interface card and connects to the campus backbone via Cisco switches and routers. At this time, all the interfaces operate at 10 Mbps.

3. AFS: Andrew File System

Andrew File System (AFS) grew out of Andrew, an academic distributed computing environment at Carnegie Mellon University. Later on, Transarc Corporation commercialized and carried out further development of AFS [5].

3.1 Overview

The file system model of AFS is primarily based on Unix model, although AFS clients have been available for Microsoft Windows NT operating system for some time.

AFS is designed for an environment consisting of a large number of untrusted client workstations with local disks. A small number of dedicated and trusted servers, collectively called *vice* run file server processes that support operations for storage and retrieval of portions of files or files in their entirety. Clients cache files and directories from the servers on their local disks. The client caching can be viewed as a logical extension of caching and buffering techniques in conventional file systems i.e buffer cache in Unix file systems. In conventional file systems, caching (buffer cache) is used to reduce disk I/O, whereas in AFS client side caching is used to reduce network traffic.

AFS uses a proprietary Rx protocol based RPC package built on top of UDP for high level client-server communication. The distinguishing features of this package are: capability of efficient bulk data transfer, authentication facilities for non-trusting clients and servers via three-phase encrypted handshake and optional use of secure communication channel using encryption.

In AFS version 2, files were cached in their entirety and consequently files larger than the cache size could not be used at all. The clients interacted with servers only during opening and closing of a file.

From AFS version 3, files are cached in chunks of 64 KB thus permitting use of files larger than the cache size. The chunk size is configurable at the boot time and is one of the tuning parameters. During reads, files are brought in the cache in chunks and writes to files are to the local copy in cache. Only on close, the file is stored back on the server. Thus changes to a file are not visible at other sites. It is possible for clients to explicitly request that changes be visible to remote clients. However, file operations such as protection changes at the directory level are immediately visible at all the sites. If a file is larger than the client cache size, the client is forced to send the chunks over to the server before the file is closed. In order to maintain the correct session semantics, the server writes the data under a different version number and updates the file only when the client closes the file. This is a potential bottleneck and operations involving file sizes greater than few hundreds of megabytes give rise to poor performance.

AFS does not support the Unix semantic model. In the Unix model, any changes to file following a write are available to all other processes sharing that file. AFS supports *file session* semantics i.e. the updates are visible only when a file is opened and changes are propagated only when the file is closed. This compromise allows clustering of reads and writes and reduces cost and complexity of the full Unix semantics model.

The file servers are dedicated machines located in physically secure locations. The clients and servers do not trust each other or the network. The AFS authentication scheme is based on a slightly modified version of MIT's Kerberos 4 network authentication service. The Kerberos4 stashes the ticket on a local file system whereas AFS stores ticket for file services in the kernel's per user data structure and is referred to as a *token*.

3.2 AFS File System

There are several differences between files residing in AFS and files in Unix file system. These are more apparant to administrators and developers than casual users. These characteristics have a direct bearing on security.

A user's AFS token is distinct from a Unix identity. When files are created in AFS, the assigned ownership is based on the token identification number and not the Unix identifier. Normally the token identification number is the same as the Unix identifier.

The AFS access is on the per directory basis. Although all the 9 permission bits for files are properly stored and a user is free to set other bits, only owner bits are used in determining the final access.

The "admin" user in an AFS cell is equivalent to "root" user on a workstation and has all the permissions in an AFS cell. Only a user with an admin token can change ownership and a group of a file.

The setuid executables work but only affect the Unix identity of a process, the AFS identity is not changed. Thus user root has no special meaning to AFS, setuid programs can be used to administer the desktop processes but not AFS. The setgid bits and sticky bits are stored correctly by AFS and used correctly by Unix.

AFS uses two entries in the per-process group table to store Process Authentication Group (PAG) information which is used to propagate this information to the child processes. A user has to use a command "unlog" to discard the tokens while logging out.

AFS does not support special devices and special devices such as sockets, pipes can not be stored with an AFS path name.

AFS allows symbolic links but permits hard links only between files in the same directory.

AFS uses the last time of file-data change (atime) for other two time stamps (ctime and mtime). For example, chmod on a file does not update any times whereas append to a file causes all the times to change.

3.3 AFS System Software

IBM Transarc supplies modifies versions of the following Unix programs:

- ftpd
- login
- rlogind
- rsh

In addition to performing normal Unix checking, these modified programs also perform AFS's Kerberos authentication steps. The modified login program authenticates to the AFS Kerberos4 and obtains a token for access to user home directory. Depending on the OS vendor and AFS version, version of rsh and rlogin transmit the token to the remote workstation.

There are two versions of inetd running on an AFS client workstation: the default the Transarc modified version. Correspondingly there are configuration files: /etc/inetd.conf for Unix inetd and /etc/inetd.conf.afs for Transarc modified inetd.

3.4 AFS Package Utility

IBM Transarc provides a utility, called package which can be used to automate the copying process from locations within AFS to local file system on a client. Some of the files thus copied at ND are listed below:

- AFC cache configuration files, afs libraries and scripts
- AFS modified binaries such as inetd_afs, /etc/inetd.conf.afs
- Sendmail files, /usr/lib/sendmail, /etc/sendmail.cf and /etc/aliases
- /etc/passwd, /etc/hosts, /etc/printers.conf, /etc/power.conf, /etc/pam.conf
- /etc/init.d/sshd, /etc/sshd_config, /etc/krb.realms, /etc/krb.conf
- /usr/sbin/tcpd, /usr/sbin/rpcbind, /etc/hosts.deny, /etc/hosts.allow

Example package file entries related to sendmail are shown below. Note that the lines have wrappped around.

```
FAQ     /usr/lib/sendmail
    /afs/nd.edu/common/custom/cluster/usr/lib/sendmail-8.10.1
    root sys 4555
FAQ     /etc/mail/sendmail.cf
    /afs/nd.edu/common/custom/cluster/etc/mail/sendmail.cf_sol7_cluster
    root bin 444
LAI     /etc/sendmail.cf        /etc/mail/sendmail.cf
F       /etc/mail/aliases       /afs/nd.edu/common/custom/cluster
    root bin 444
LAI     /etc/aliases            /etc/mail/aliases
```

The names of files, their modes and ownership information is stored in a package file in AFS. The name of this configuration file is stored locally in the file /usr/vice/etc/config/package.options. The package utility is always run on reboot and can be run manually by root. If the timestamp on any of these files is new, the package utility is followed by a reboot.

This is a great administrative convenience as it is possible to propagate versions upgrades merely by changing one configuration file.

3.5 Implications of root compromise on an AFS Client

Any misconfigurations, security lapses, tardy patching practices may lead to root access on a client workstation. With root access on an AFS client workstation:

- it is possible to to steal tokens of currently logged on users or of those users who have logged out but did not discard their tokens using unlog command and access their files.
- it is possible, although not easy or straightforward, to read all the data in client cache.
- it is possible to trojan the binaries to capture passwords.
- run the network interface in promiscuous mode and sniff the traffic
- use the workstation as a staging area for attacking other sites and use local disk for storage.

Since all the workstations in the cluster operate in switched environment the last possibility is reduced but not eliminated [3].

Until about a year ago, the GNU finger was installed on the cluster (one central machine serving as data collection host) and was for the most part a convenience for many for locating their friends and project partners. However due to some harassment and stalking incidents, this service was shut off and there are no plans to make it available again.

4. Installation Procedure

The version of the OS on the workstations is for the most part dictated by the application software and the availability of the AFS client. It is typically about 6 months to a year behind the latest version available from Sun. For example, at this time the Solaris 8 AFS client is in beta.

The following sections describe the installation procedure. The installation of the OS, AFS, customization and creation of the master disk is performed by the author. The actual upgrade is performed by the cluster group.

4.1 Install Solaris 7

The application software used in academic curricula is diverse and consequently the complete version of the OS is installed on the workstations. Severe time/resource constraints preclude a systematic pruning of the number of packages installed. Because it is difficult to predict the behavior of various applications in 64 bit mode and some applications can be run only in 64 bit mode, both 32 and 64 bit OS packages are installed. A system could be configured to boot in 32 or 64 bit via PROM boot-file setting.

4.2 Install AFS

After the OS installation is complete, a tarball containing the afs libs and scripts is downloaded from AFS via ftp and AFS is installed. This installation script and other scripts are home grown Bourne shell scripts, together about 1500 lines. The install script does the following:

- Check OS version, root invocation
- Add entry for afs in /etc/sys_to_sysnum file, reboot and reinvoke the script. If an entry exists continue with installation.
- Check if the root has a password, if not ask to set it
- Disable keyserv, automounter, snmpdx, dmi daemons
- Create/Modify /etc/defaultrouter, /etc/resolv.conf, /etc/nsswitch.conf files with proper modes and permissions.
- Create AFS files/directories, load AFS module, start afs
- Create /usr/afsws, /usr/local, /opt links to the AFS world.
- Modify /etc/inetd.conf and create /etc/inetd.conf_afs Replace some OS binaries with AFS versions (login, passwd etc.)
- Replace OS /etc/pam.conf with a version containing AFS entries
- (*) Run "Enhance Security" script
- Replace OS version of Sendmail with the one in AFS
- Run the package file to download latest versions of files included in the package configuration file
- Reboot the system

The "Enhance Security" script has the following steps:

- Turn off execution off the stack
- Turn off ToolTalk, KCMS, Font Server, CacheFS, kerbd, sadmind daemons
- Turn off sprayd, echo, discard, daytime, chargen services
- Turn off rquota, rusers, walld, time, uucp, talk, comsat, name
- Install TCP wrapper daemon and wrap telnetd, rlogind, rexecd and ftpd
- Turn off printer, rstatd, cmsd daemons
- Install ssh and generate a host keys

The script also installs OpenGL packages, AFS aware utilities like xlock, runs catman command on local man directories, adds tcsh to /etc/shells, renames /usr/openwin/bin/fs which clashes with AFS fs suite of commands.

4.3 Customization

This install script is run by non-OIT staff members as well to install AFS clients on their departmental workstations. The script is "non-fascist" in the sense that it provides choices, for example with dtlogin vs. command line login. For the installation of the cluster disk, the most restrictive options are chosen.

After running this script, the OS is patched with the latest security and recommended patches. The patches and a short script to install the patches is stored in AFS.

4.4 Create Master Disks

After testing and analysis for security vulnerabilities, 3 to 5 master disks are created using the command dd and some minor modifications. Master disk, housed in external enclosures contains the image of the internal disk plus a script to customize the workstation. The script uses data from a file contains the hostids, correponding hostnames, IP numbers and the default router entry.

4.5 Uprade Workstations

The master disks are handed over to the Cluster group who upgrade the workstations. The installation process takes about 15/20 minutes per workstation and has the following steps:

- Power down the workstation
- Connect the disk drive to the workstation
- Boot off the external disk
- Run the script
- Power down the workstation, disconnect the disk drive
- Boot the workstation

5. Security Analysis of Prototype Workstation

The prototype machine is located in the author's office. For the most part it is physically secure. During preparation of the prototype there are windows of vulnerability.

The prototype machine is vulnerable to network attacks during OS install. This was easily avoided by disconnecting the machine from the network.

The prototype machine is vulnerable to network attacks because of sequence of download of AFS tarball, AFS client installation and patching. The author is aware of the possibility of creating CDs with the latest patches, patching and then installing AFS. At this time this is an acceptable risk.

The prototype machine is different than the cluster workstations in physical security and the vulnerability during the installation.

6. Security Analysis of Cluster Workstation

This section is divided into 7 different categories: physical, host based, third party software, network services, administrative practices, backups procedures and disaster recovery. Most of the Solaris security related information was derived from Sun Microsystem books [7] and [8].

6.1 Physical

The cluster machines are located publicly accessible labs, some of which are open 24 hours and are monitored only between 8:00 am to 10:00 pm on weekdays. The lab doors have combination locks and all the students/staff/faculty can get the combination by asking. The campus is located in a fair sized town South Bend (IN) but is fairly isolated and people not associated with Notre Dame are rarely seen on campus except during special events.

Vulnerability: Theft

Precaution: CPUs have a fiber optic cable running through a fixture at at the back. The campus security is alerted if the cable is disconnected or broken.

6.2 Host based

Vulnerability: Unauthorized booting in single user mode

Precaution: There is a Firmware/PROM password to prevent booting in single user mode.

Risk:It is possible to corrupt the filesystem by repeated power cycling to get root access.

Vulnerability: It is possible to deny services by filling up /tmp

Precaution: Limit the size of /tmp to 1 GB, the relevant /etc/vfstab entry is:

```
swap    -       /tmp    tmpfs   -       yes     size=1000m
```

All the unnecessary entries were cleaned out from /var/spool/cron/crontabs. The policy decision about whether to disallow user access to crontab has not been made. User cron jobs run only with Unix access rights and not with AFS credentials, consequently can not write to user home directories. Users get around this by making some directories writable to campus users. Because most of the workstations are idle after midnight, user convenience has taken precedence over security considerations.

Password aging for AFS passwords is not turned on. The restrictions on password are the standard ones: 6 to 8 characters long, with at least one non-alphabetic character. Some users have not changed their passwords for years. There is an AFS version of Crack [2] which works against AFS kasserver (authentication) database. The AFS kaserver machines are in a secure central location, with all non-essential network services turned off with no remote logins without ssh etc. It is not easy to get a copy of kasserver database. However if a copy is obtained, the AFS crack runs much faster against the database than traditional Unix encrypted password, because in string to key conversion, no salt is used.

Because of tight schedule, no audit was performed on the OS related files on the local disk. Tools like ASET and Fix-mode were not run to close unnecessary permissions on various files and directories. Permissions on log files, /dev/term/? entries etc. were however checked.

6.3 Risks from Third Party Software

All the third party software without exception is installed in the AFS file system as an ordinary user without "admin" access. All the installations are performed from workstations on staff offices.

6.4 Network Services

Vulnerability: Buffer overflows because execution off the stack

Precaution: Turn off execution off-stack, relevant entry from /etc/system

```
set noexec_user_stack=1
set noexec_user_stack\_log=1
```

The entries in /etc/inetd.conf are:

```
ftp     stream  tcp     nowait  root    /usr/sbin/tcpd  in.ftpd
telnet  stream  tcp     nowait  root    /usr/sbin/tcpd  in.telnetd
login   stream  tcp     nowait  root    /usr/sbin/tcpd  in.rlogind
exec    stream  tcp     nowait  root    /usr/sbin/tcpd  in.rexecd
dtspc stream tcp nowait root /usr/dt/bin/dtspcd /usr/dt/bin/dtspcd
```

It is not entirely clear whether dtspc needs to be running at all. Some users have reported problems in managing their CDE environment when it was shut off. The author plans to investigate this further.

The entries in /etc/inetd.conf.afs are:

```
shell stream  tcp   nowait  root  /usr/sbin/tcpd   in.rshd
ta-rauth  stream  tcp  nowait  root  internal         ta-rauth
```

Vulnerability: Vulnerable network services

Precautions: Shut off ToolTalk, KCMS, Font Server, CacheFS, Kerbd, sadmind, sprayd, echo, discard, daytime, chargen, rquota, rusers, walld, time, uucp, talk, comsat, name

Vulnerability: Telnet and ftp passwords in clear text

Precautions: Due to demands from PC/Mac users these services can not be shut off, however off-campus access to these has been shut off using TCP wrappers. Installed ssh-1.2.27, which has AFS support

Vulnerability: Sun Portmapper (rpcbind)

Precaution: Install Wietse Venema's rpcbind_2.1 with libwrap support. Shut off access to off-campus probes.

The entries in /etc/hosts.allow are:

```
sendmail: .nd.edu, 127.0.0.1
```

```
in.telnetd: .nd.edu
in.ftpd: .nd.edu
in.rlogind: .nd.edu
in.rshd: .nd.edu
rpcbind: 129.74.0.0/255.255.0.0
```

The sendmail entry is explained in the Sendmail section.

The entries in /etc/hosts.deny are:

```
ALL: ALL: banners /usr/banner
```

Vulnerability: Trojaned Open Source software

Precaution: Install pgp-6.5.2 for signature verification.

Vulnerability: xhost based X display permissions

Recommendation: Use ssh and not use xhost +, educate users.

Vulnerability: root logins without audit trail

Precaution: Disallow root logins (unset CONSOLE variable in /etc/default/login file)

Vulnerability: ssh keys reside in AFS user home directories

Recommendation: Acceptable risk

The following checks were performed. Although the author has casually perused the document by Voeckler in TCP/IP stack tuning, the author does not understand the tuning of TCP/IP stacks well enough to comment on the default settings of network devices.

- ndd /dev/ip ip_forwarding returns 0, OK
- ndd /dev/ip ip_send_redirects returns 1, needs to be 0
- ndd /dev/ip ip_forward_src_routed returns 1, needs to be 0
- The file /etc/ftpusers does not exist, needs to be created
- TCP_STRONG_ISS in /etc/default/inetinit set to 1, needs to be 2

An nmap scan from another workstation shows the following:

```
Starting nmap V. 2.53 by fyodor@insecure.org ( www.insecure.org/nmap/ )
No tcp,udp, or ICMP scantype specified, assuming vanilla tcp connect() scan. Use -sP if you really don't want to portscan (and just want to see what
Host <deleted> appears to be up ... good.
Initiating TCP connect() scan against <deleted>
Adding TCP port 601 (state open).
Adding TCP port 25 (state open).
Adding TCP port 21 (state open).
Adding TCP port 4045 (state open).
Adding TCP port 32771 (state open).
Adding TCP port 6112 (state open).
Adding TCP port 587 (state open).
Adding TCP port 22 (state open).
Adding TCP port 514 (state open).
Adding TCP port 512 (state open).
Adding TCP port 111 (state open).
Adding TCP port 23 (state open).
Adding TCP port 513 (state open).
The TCP connect scan took 1 second to scan 1523 ports.
Interesting ports on install.cc.nd.edu (129.74.36.110):
(The 1510 ports scanned but not shown below are in state: closed)
Port       State       Service
21/tcp     open        ftp
22/tcp     open        ssh
23/tcp     open        telnet
25/tcp     open        smtp
111/tcp    open        sunrpc
512/tcp    open        exec
513/tcp    open        login
514/tcp    open        shell
587/tcp    open        submission
601/tcp    open        unknown
4045/tcp   open        lockd
6112/tcp   open        dtspc
32771/tcp  open        sometimes-rpc5
```

The port 601 is associated with AFS authentication service ta-rauth, The SMTP services are offered on ports 25 and 587. As remarked earlier telnet, ftp, login, shell, exec are available only for on-campus access.

6.5 Sendmail

Because of a long history of vulnerabilities Sendmail deserves its own section. Currently the cluster workstations run Sendmail 8.10.1 [4]. The Sendmail 8.10.2 has a workaround for broken Linux setuid() implementation and adds more vigilance around some system calls. Sendmail 8.11.0 [4] fixes vulnerabilities due to broken snprintf() implementations. The test program test/t_snprintf.c was compiled on Solaris 7 and it reports that snprintf works properly. The sendmail binary is installed setuid root.

All the cluster workstation have MX aliased to a centrally located machine called mailspool.helios.nd.edu. It receives mail on behalf of all the clients and as remarked earlier, all the clients NFS mount the /var/mail directory.

The following were checked [6]:

- /,/usr,/usr/lib are owned by root. The permission bits on /usr are 775, should be 755?

- /usr/bin/newaliases is a link to /usr/lib/sendmail.
- /var/spool/mqueue is 750 and owned by root, should be 700.
- EXPN, VRFY and DEBUG are turned off.
- decode alias is set to root to monitor attempts to use this alias.
- The sendmail runs a daemon with queue flush set at 15 minute interval. It is run as a daemon because many of AIX clients decline to read MX records. It may be better to run sendmail through cron rather than as a daemon.

The m4 file from which the sendmail configuration file generated is as follows:

```
divert(-1)
#
# ND cf config for solaris 2.x
#

divert(0)dnl
include(`../m4/cf.m4')
VERSIONID(`@(#)nd_cluster_nullclient.mc 8.10.1 5/2/00')
OSTYPE(solaris2)dnl
FEATURE(always_add_domain)dnl
FEATURE(redirect)dnl
undefine(`ALIAS_FILE')
define(`MAIL_HUB', `mailspool.helios.nd.edu')
define(`SMART_HOST', `mailspool.helios.nd.edu')
define(`confFORWARD_PATH', `')
define(`confPRIVACY_FLAGS', `authwarnings,restrictmailq,restrictqrun,noexpn,novrfy')dnl
define(`confME_TOO')dnl
define(`confCF_VERSION', `ND-cluster')dnl
MAILER(smtp)
```

All local delivery is forwarded to the central server for delivery (defined by MAIL_HUB). The non-locally deliverable mail is delivered to the central machine (defined by SMART_HOST).

6.6 Administrative Practices

From the security perspective, this is by far the weakest area. The reasons for this situation are complex and beyond the scope of this report. In this web version of the report, this section is truncated.

Every morning around 9:00 a.m. rdist from a centrally located host distributes a new /etc/passwd file. After distribution a short script preserves the information about root in the file /etc/shadow, and regenerates the /etc/shadow file.

Recommendations: Develop clear policies and procedures and increase security awareness.

6.7 Backup Procedures

Since all the application software and user home directories reside in the AFS, backups of any individual client workstations are not necessary and not performed. If a disk is trashed or a machine needs to be rebuilt, with a master disk it takes only 15 to 20 minutes. There are between 3 to 5 master disks, with at least one in the author's office, there is no off-site storage. In the worst case scenario, the disk can be built from scratch in about 4 to 6 hours.

6.8 Disaster Recovery

In the event a workstation is broken into, after some rudimentary forensic analysis to identify vulnerabilities, attack signatures and if applicable determining the requisite patches, the OS is reinstalled from scratch using a master disk.

A tripwire database of all the local files was created using an older version. The latest freely version tw_ASR_1.3.1_src from tripewire.com is built and being tested.

The Unix Forensics session recommended creation of analysis CDs with static versions of important tools. At this time this is not done.

6.9 Summary

The weakest element is without question poor administrative practices.

The major technical weakness in securing the workstation is lack of careful audit of the basic OS installation. Tools such as ASET, Fix-modes, SARA, Nessus etc. needs to be used to probe the vulnerabilities.

Bibliography

1
http://www.transarc.com.

2
ftp://export.acs.cmu.edu/pub/afs-tools/afscrack-1.1.tar.z, 2000.

3
http://www.monkey.org/dugsong/dsniff, 2000.

4
Release notes, sendmail 8.10.2/8.11.0 distribution, 2000.

5
Richard Campbell.
Managing AFS, Andrew File System.
Prentice-Hall, 1998.

6
Bryan Costales, Eric Allman, and Neil Rickert.
sendmail.

O'Reilly and Associates, Inc., 1994.

7

Peter H. Gregory.
SOLARIS Security.
Sun Microsystem Press, 2000.

8

Janice Windsor.
Solaris Advanced System Administrator's Guide.
Sun Microsystem Press, 1998.

About this document ...

Security Analysis of AFS Client Workstations

This document was generated using the **LaTeX2HTML** translator Version 98.1p1 release (March 2nd, 1998)

The command line arguments were:
latex2html `-split 0 -ascii_mode -show_section_numbers milind_saraph.tex`.

The translation was initiated by Milind Saraph on 2000-08-21

[Next Group] [Up] [Previous Group]

Milind Saraph
2000-08-21