- Trusted X server, including all server operations, with the Motif window manager.
- Trusted Motif window manager, including label display and trusted paths.

Four specific security categories are addressed:

1) Associating labels with all window system objects. This includes a set of client idenfications and labels, as well as labels for Atom Names, Properties, Colormaps, Pixmaps, and Drawables. For any Drawable (a window or a pixmap), the trusted X server uses three labels:

Sensitivity label (normally set equal to the parenting client sensitivity, but modifiable)

Information label (dynamically modified with window operations)

Input information label (which determines the acceptable input events sent to the window)

2) Access control and operating restrictions for trusted clients. There are two specific controls applied;

Certain window objects are accessible only to their single creating client

Shared objects are controlled by explicit, mandatory control checking.

3) Protocol extensions and modifications for the trusted window manager. The window manager operations are modified in five ways:

Cut/paste operations are governed by authentication clients; data transferred is label verified.

Authentication procedures are given to the trusted window manager, including starting new sessions.

Support for window-based trusted paths, including reserved screen areas.

Support for label modification (clear or reset operations) processes.

Support for inactive session or timeout processes from a configurable logoff event.

4) Auditing events and security concerns. The trusted X server and trusted window manager create and maintain audit records for these events:

Creation and destruction of window system objects

Explicit changes to window labels

Implicit changes to information labels resulting from floats

Access denials resulting from access control checking

Access denials resulting from lack of privilege

Client use of privilege

Server reinitializations

1.11.4.6 Using the ViewStation’s xhost Client

If you have an R5 Xdm client with an Xsetup file, you can solve authentication problems by using the **xhost** client, a local process running on the HDS ViewStation. This solution has the HDS ViewStation grant access to other hosts by means of the **xhost** program. Because the ViewStation runs this **xhost** program at Xdm startup, it sets this permission as Xdm starts up.

In the Xdm directory on your host machine, edit the Xsetup file (on a Sun system - */usr/openwin/lib/xdm/Xsetup*; on an HP system - */usr/lib/X11/vue/vuelogin / Xsetup*; on SCO systems - */usr/lib/X11/scologin / Xsetup*; etc.). This Xsetup file may not exist and you may have to create it. If it is not present, you must also enter this Xsetup file name in the xdm configuration file, xdm-config. Refer to Section 3.3.2.

This sample script checks for HDS ViewStations among displays requesting Xdm management, then sets “xhost +”, which extends access permission to all hosts.

```
vendr=‘xdpyinfo | grep Vendor | grep HDS‘
if [ -n ”$vendr” ]
then
# this assumes that the ViewStation display is registered in
# the /etc/hosts file or with NIS on the host
                disp=‘echo $DISPLAY | sed ’s/:/ /’ | awk ’{
print $1 }’‘
                rsh $disp xhost +
fi
```

Be careful with the syntax in this file; make adjustments appropriate for your host machine (such as “remsh” for HP hosts).

The ViewStation’s initial login to the Xdm host (which creates the Cookie authorization) then has permissions extended to all other hosts.

1.11.5 CMW Security

The HDS ViewStation supports Compartmentalized Workstation (CMW) security, in which applications and processes are controlled by a "Trusted Client", typically a window manager. This trusted client maintains labels containing permission levels for all windows, X atoms, and pieces of data for all clients and servers. These labels are examined with each transaction and a detailed audit trail is maintained. This CMW security meets U.S. Government B1 level security, using the NSA rating procedure.

1.11.5.1 CMW Security Features

This section summarizes the compartmentalized workstation secure window interface for the HDS ViewStation terminal. This includes modifications to the server software based on X11R4. These comprise the TCB (Trusted Computing Base). There are two main features of CMW security:

1.11.4.4 Magic Cookie Security on the ViewStation

The HDS ViewStation controls its response to Magic Cookie requests by the "Host Access Control" field in the XDMCP menu of Setup Mode. This permits you to enable or disable this security feature on the ViewStation. Note that this is turned ON in its default configuration.

1.11.4.5 Problems with Magic Cookie Authentication

Sometimes a problem arises with Magic Cookie authentication. Its symptom is that the user can log on with Xdm and use processes on the Xdm host but cannot start processes on other hosts. There are two difficulties observed.

- Authentication is On by default with R5; users may not be aware of the authentication process. Typically, the problem doesn't show itself until the user tries to log onto a host other than the Xdm host and cannot do so.
- A faulty "Magic Cookie" is created by some Xdm clients

In both of these cases, there are ways to disable or to extend the authentication so your system can be accessible to all users. These are important considerations and need careful security control.

Extending User Authentication

Most systems have an MIT resource for the login client that permits disabling the access control. This resource is available in both R4 and R5 versions. In your */usr/lib/X11/xdm/Xresources* file, set the resource:

```
xlogin*allowAccess: true
```

This allows access to the session for all users, thus effectively disabling Magic Cookie authentication.

You can also set this resource with the key sequence *Ctrl*-+ (some systems have not implemented this key translation to set the allowAccess resource from the keyboard). You can use the *Ctrl*-+ key sequence at any time, including when the Xdm login window is present. When you press the *Ctrl*-+ sequence, you will notice the login banner change to "This is an unsecure session", which means that the authentication feature is disabled. Note that *Ctrl*-+ toggles the allowAccess resource, so if it was initially Off, this can turn it On. In this way, the user can select the authentication he wants for the system.

Extending Magic Cookie Authentication

We have observed that some versions of Xdm (a host-based client) do not work correctly and create an invalid Magic-Cookie. This means that the user can only gain access to the original Xdm host. Other hosts will have access denied and will report authentication and permission errors. This condition arises because the host's Xdm process generates an incorrect Cookie (typically all F's).

If you have this problem, you can explicitly extend authentication to one or more hosts. This is the intended method of operation, so user security is controlled by the Magic Cookie. In a practical sense, extending authentication to all hosts is the same as disabling the authentication.

equivalency of his user name for other hosts. You may also have to modify this file. The syntax for this file is similar but varies with different operating systems.

The ViewStation's individual rhosts file is discussed in Section 1.11.2.

1.11.4 Host Access Control Security

The host-based security described above can also be extended to a user-based security procedure, called MIT-MAGIC-COOKIE-1, which is supported by the HDS ViewStation. This security procedure extends the security of the user's account and password to all clients on the system. The procedure is simple and almost invisible to the user.

Note that for some systems (such as OpenWindows), this security is on by default.

1.11.4.1 Host Access Control from Setup Mode

A new resource has been added to permit the ViewStation to use Access Control security when you are not using Xdmcp. It uses the first remote client to connect as its Access Control.

The resource is in the config.syntax.fx file. The default is disabled.

```
HDSsetup.default.AccessControl: [ DISABLED | ENABLED ]
```

1.11.4.2 Using Magic Cookie Security

When a session is started with Xdm, a special code (the Magic Cookie) is written to the .Xauthority file in the user's home directory. This .Xauthority file has restricted permissions (-rw-------) so only the original user can read it. Before any new clients can start, they must read this file to obtain the Magic Cookie to be able to start their process. If they aren't the original user, they can't read the file and access is denied. The key to Magic Cookie security is the integrity of the user's account and password. If anyone can read the user's .Xauthority file, then this security mechanism is useless.

1.11.4.3 Starting Magic Cookie Security in Xdm

This security is started in the Xdm configuration file, xdm-config, with two resources:

```
DisplayManager._0.authorize: true
DisplayManager*authorize: true
```

The first resource turns on authorization for the local display and the second turns it on for for all other displays. You can disable authorization by setting these to "false" and restarting Xdm. This is the proper way to disable authorization for your Xdm process.

There is also a way to disable it by allowing authorization for all users. This is discussed in the sections below. More details on Xdm operation and setup are found in Section 3.3.2.

1.11.3 Permission for Remote Shell Commands

The process of starting the local window manager from a remote host uses the **rsh** (or **rcmd** on SCO hosts or **remsh** on HP hosts) command. This process establishes privileges for the ViewStation to use the remote host's permissions to start other remote processes on other network devices. These permissions are governed by the *hosts.equiv* and *.rhosts* files, discussed below.

With these things in place, the ViewStation is a trusted machine, that is, if a remote shell command comes from a particular host to the ViewStation, it will permit remote shell processes only back to that host or hosts specified in the permission files. The initial user logon from the ViewStation to the host sets the username, password, environment, and permissions. These permissions are extended to the ViewStation by the remote shell process, and any further extension of those privileges is governed by the *hosts.equiv* file. In this way system security is preserved at the same level for the ViewStation as an X terminal and any host or workstation on the network. The ViewStation is no more vulnerable (or any less vulnerable) than any other device on the network.

1.11.3.1 Using the */etc/hosts.equiv* File

The most common security permission is set in the */etc/hosts.equiv* file, which is a simple list of hostnames which have permissions equivalent to the host. Generally, you must add the ViewStation's hostname to this file on the remote host you will be using.

Syntax for this *hosts.equiv* file varies with different operating systems. Adding names to this file is not done automatically (some operating systems, like Sun OS, do permit the addition of a plus sign "+" to this file, which indicates permission for automatic addition of host names), but in general, you must explicitly add the ViewStation's name to this file. The following examples are suggestive of the contents of this file. Check your own man pages for correct syntax on your system.

```
# hosts.equiv syntax examples
#
# host1 userA                 this allows userA access to host1
# + host1                     this allows all users access to host1
# host1 -userA                this denies userA access to host1
# - host1                     this denies all users access to host1
# -@groupA                    this denies groupA users access to hosts
# +@groupA  +@groupB          this allows groupB users access to
                              groupA hosts
```

1.11.3.2 Using the .rhosts File on the Host

Another location that may have permissions set, or prohibited, is the *.rhosts* file in the user's home directory. This file allows the user to specify permissions and its Setup Mode or its NVM. The NVM password and the bootptab passwords must be the same, and only the encrypted password appears in the configuration file. Since the bootp protocol and bootptab are controlled exclusively by the superuser, they may be protected or hidden.

Note that special characters, like spaces, can be used in encrypted passwords if they are enclosed in double quotes "". This ViewStation feature is present in boot prom versions 3.0 and higher.

With HDSware 3.0.3 and higher (you must use both boot prom code and server code at the V3.0.3 level), the ViewStation accepts any password in the configuration file as either plain text or encrypted text.

1.11.2 ViewStation rhosts Security

The ViewStation uses an internal rhosts file to configure the rsh and rlogin daemons and control access to the ViewStation. By default, this security is not used. This feature is available with HDSware version 3.2 and later.

The ViewStation's rhost security is set up by entering an rhosts file name in Setup Mode on the Remote Hosts menu or with the resource:

```
HDSsetup.rhosts.file: <filename>
```

in the configuration file. If no file is specified or the file is missing, rhosts security is not used. If the file is present but empty, no access to the ViewStation is allowed.

1.11.2.1 Setting Up an rhosts File on the ViewStation

An rhosts file is specific for an individual ViewStation. It can have any name or be in any location, but typical filenames might include the ViewStation name, such as "gregVS.rhosts" and might be located in the HDSware configuration directory (*/HDSware/hds-fx/config/*). These files should be writeable by root only.

The file contains records in this format (with white space separation):

remote_host_name   remote_user_name

Each record should be on a separate line. The record gives permission for a host and user to use rsh and rlogin for communication to the ViewStation. The remote_host field accepts a DNS name, an IP address, a list of hosts (comma separated), or a "*" wildcard for all hosts. If the DNS name does not include an "." in its entry, the DNS suffix will be appended to the name when it is added to the database. The remote_user field accepts any valid username or "*" for any user.

1.11.2.2 ViewStation Remote Access Reporting

The ViewStation reports each host and user access attempting to connect to it with messages in its Console Window. Some sample messages are:

```
rshd_io: accepted rsh by gregh@hdssun1.hds.com
rshd_io: accepted rlogin by gregh@hdssun1.hds.com
rshd_io: rejected rsh by gregh@hdssun5.hds.com
rshd_io: rejected rlogin by gregh@hdssun5.hds.com
```

1.11.1.6 Passwords in Bootp

If you use Bootp with your first time startup to load the configuration file, you must use TFTP as the file transfer protocol (or anonymous FTP with its username and password to load the file), and since TFTP requires a world-readable file, the password in the configuration file must be encrypted to preserve its integrity. The password in the bootp table itself can appear in clear text since this file belongs to the root user and can be hidden or read-protected. In this way, the level of system security of the root user is extended to each ViewStation.

If you use the ViewStation to load the configuration file directly , you can use the FTP or NFS protocols for the file transfer. You can also read-protect the file, so you can use the password in clear text.

1.11.1.7 Password Encryption with hdscrypt

The ViewStation has an decryption routine built into its server (based on a public-domain routine called "crypt" which is included with MicroEMACS written and copyrighted by Dana L. Hoggett; this program, renamed "hdscrypt" to avoid confusion with other encryption programs, can be obtained directly from HDS). This encryption program allows you to encrypt your password and then have the ViewStation decrypt it for comparison with the Setup Mode password(s) as needed.

To use Bootp to load a configuration file at initial startup, you must use TFTP (or anonymous FTP with a username and password). The configuration file itself must have read permission for TFTP (and the rest of the world) in order to be moved. If the Setup Mode password is used in the configuration file, it must be encrypted to preserve its integrity.

Using the hdscrypt program is simple. When you run the hdscrypt program, it requests a "key", that is, your user password, which serves as the basis for the encryption scheme. In this example:

```
hdssun12: hdscrypt

Enter Key rembrandt

rembrandt

Encrypts to: [IXWnI‘&}z]
```

the Key entered is "rembrandt", which is the user’s password. You then re-enter the password and it is encrypted against this Key. The encrypted password is then displayed in brackets. You can encrypt other passwords (such as the FTP password) using this same key, but note that all these passwords must be used in the same configuration file, since the decryption is done against the same key.

This password (key) is also entered in the T128 field in the bootptab file as plain text. The encrypted password "IXWnl’&}z" is the entry you make in the configuration file.

When the configuration file is loaded into the ViewStation, the ViewStation server code (which also runs the hdscrypt program) examines the entry and decrypts it into "rembrandt", which it then compares to the password entered by the user in attempting to start the client. The password is entered by using the ’password’ command in the HDS Console Window.

Almost all Console shell commands can be password protected, according to the following list.

You must set passwdRequired for these commands:

```
arp            cat            date           ether
hdsmemory      help           hwconfig       list
log            logpg          ls             more
netstat        nvm            opendev        ports
ps             pwd            rehash         version
```

The passwdRequired resource is set by default for these commands:

```
db             dw             fdisk          flash
kill           reboot         restart
```

These commands cannot be used with the passwdRequired resource since they are integral parts of the shell itself:

```
cd             echo           env            exit
password       setenv         unsetenv
```

You can set this resource just like with any other HDSperx client, such as:

```
hds.client.list.passwdRequired: true
hds.client.ps.passwdRequired: true
hds.client.netstat.passwdRequired: true
hds.client.opendev.passwdRequired: true
```

1.11.1.5 Password Security

The ViewStation’s password is not displayed on any screen. It must be entered before it is changed, either locally or remotely. The password can be encrypted if it needs to be transmitted over the network; the ViewStation will read and decrypt the password before it does its password verification.

If the system administrator sets a password in the configuration file and then sets that file to be saved in NVM, the user cannot change the password unless he knows it and has access to Setup Mode and the configuration file. The configuration password and the NVM (Setup Mode) password must always match. If the configuration password is loaded first, it is used as the current Setup Mode password; if the NVM password is loaded first, the configuration file (with all its settings) is discarded if its password doesn’t match the NVM password. Since many Setup Mode fields, such as fontpaths, IP addresses, privileged Console window commands, etc. are protected by this Setup Mode password, important ViewStation settings are secure.

Handling the password on the network system needs some care.

1.11.1.2 Setup Mode Password Protection

The password protects sensitive fields in Setup Mode, such as IP information, fontpaths, etc. These fields cannot be edited except by entering the password.

Further protection is available by using resources in the configuration file. The resources:

```
HDSsetup.password.is_global: yes
HDSsetup.hide_icon: yes
```

let you make all of the fields in Setup Mode and all local clients password protected and then hide the Setup Mode icon (the icon can be restored by hitting the *Alt-SysRq* key). Refer also to Section 1.5.1 of this manual.

Further, entries in the configuration file override entries in Setup Mode and NVM. In this way, the system administrator can make settings for the ViewStation that the user cannot change.

1.11.1.3 Console Window Password Protection

Commands in the Console Window that control the ViewStation’s internal operation are divided into two groups, with sensitive commands being password protected.

This is the summary of Console Window commands.

```
Summary of valid commands:
arp            cat            cd             date
env            echo           ether          exit
hdsmemory      help           hwconfig       list
log            logpg          ls             more
netstat        nvm            opendev        password
ports          ps             pwd            rehash
setenv         unsetenv       version
Summary of privileged commands:
db             dw             fdisk          flash
kill           reboot         restart
```

You must enter the password to use any privileged command, including password change.

1.11.1.4 HDSperx Password Protection

The HDS ViewStation’s local clients can be individually password protected by means of client attributes in the clients.config file. These clients include the HDSperx clients and also many shell commands.

Each client has an entry in the clients.config file. The “passwdRequired” attribute can be added to these entries to make the client password protected.

If the passwdRequired attribute is set to true, the client can only be run if the ViewStation user has been able to present the ViewStation’s password before

HDS ViewStation: Security Considerations

This chapter discusses the ViewStation's internal security, security for remote shell processes, and some system security techniques like host access. There is also a summary of the ViewStation's optional CMW security.

Section 1.0 - HDS ViewStation

1.11 HDS ViewStation Security Considerations

There are a number of important issues regarding security on an X terminal. Since the HDS ViewStation has a Unix micro-kernel as its system architecture, considerations about ViewStation access and client operation also arise.

1.11.1 ViewStation Passwords

The HDS ViewStation has a password that can be set in the User Preferences menu of Setup Mode. The password must be entered any time a protected field is edited. Only one entry of the password is needed for each Setup Mode or Console Window session. When Setup Mode or the Console Window is exited, the password must be entered again to edit protected fields.

A password entered in a configuration file, or from a bootp response is compared to the password stored in the ViewStation's non-volatile memory. The passwords must match or the operation fails.

1.11.1.1 Changing the ViewStation Password

The ViewStation's password can be changed locally by editing the field in Setup Mode (User Preference menu), or remotely by using the password command.

Using a remote login to the ViewStation, the Console Window lets you enter the password command with the change option:

```
password change
```

which then lets you change the ViewStation's internal password. This operation is password protected.