Eagle Network Security Management System

User's Guide

Version 2.2.1

Raptor Systems Incorporated

69 Hickory Drive
Waltham, Massachusetts 02254

(617)–487-7700
(617)–487–6755 (fax)

Copyright restrictions

Patents are pending on the **Eagle Network Security Management System**, **Eaglet Subnet Partitioner**, **EtherGuard**, and the **Dynamic Activity Monitor**.

Contents

List of Figures

List of Tables

Preface

With the spread of *TCP/IP*-based networks and their connectivity to the world-wide Internet, security (preventing unauthorized use of computer resources) and authentication (knowing who is using a resource) are becoming increasingly important. The **Eagle Network Security Management System** provides a way to control access by destination and source machines, and to limit the services provided.

A List of Helpful Books (and Other Information Sources)

There exists a large body of literature covering networks and security. Listed below are some of the best we have found. Each of these documents is available via anonymous file transfer from various sites on the Internet. We are grateful to the authors who prepared these documents and made them freely available. In that spirit, we will gladly provide machine-readable copies of any of the **Raptor** documents upon request.

- Guidelines for the Secure Operation of the Internet
- Coping with the Threat of Computer Security Incidents
- Site Security Handbook (RFC 1244)
- Network Reading List
- Improving the Security of Your UNIX System
- Establishing a Computer Security Incident Response Capability
- Introduction to the Internet Protocols
- Introduction to the Administration of an Internet-based LAN
- Glossary of Network Terms

There are other documents which you might be interested to read, but copyright restrictions prevent us from distributing them freely. These include:

- Security Problems in the TCP/IP Protocol Suite
- Introduction to Computer Security for Lawyers
- Design of a Secure Internet Gateway
- Foiling the Cracker

These are available from many sites on the internet via anonymous ftp. In particular, through the courtesy of John Scoggin at Delmarva Power and Light, one may copy a variety of security-related documents from:

Machine: ftp.delmarva.com
IP Address: 138.39.7.10

For good references on the fundamentals of Unix security, we recommend *Practical Unix Security* by Spafford and Garfinkel, published by O'Reilly & Associates as one of their 'Nutshell Handbooks', and *Unix System Security* by David A. Curry, published by Addison Wesley. Appendix E of *Practical Unix Security* contains an extensive bibliography of computer security books.

The 'Rainbow' series is a collection of U.S. Government volumes which describes functional and assurance requirements for computer systems, trusted databases, and assembling systems out of trusted components, and network systems. You may request one complimentary copy of the 'Rainbow' series of computer security manuals by calling (410) 766-8729 or (410) 688-8742 or by writing:

Department of Defense
National Security Agency
ATTN: S332
9800 Savage Road
Ft George G. Meade, MD 20755-6000

Finally you might want to review *Federal Criteria for Information Technology Security, Volumes 1 & 2.* It is currently in draft form and is available via `ftp` from `ftp.delmarva.com` as `/pub/security/fcvol`.

Mailing List

Through the courtesy of John Scoggin at Delmarva Power and Light, there is an electronic mail discussion group about the **Eagle** family of products. You can subscribe to this group by sending mail to

raptor-request@delmarva.com

with your electronic mail address and a request to subscribe.

Support

For technical support, call our offices. Our U.S. mail address and phone numbers are as follows:

Raptor Systems Incorporated
3422 Old Capitol Trail, Suite 3331
Wilmington, Delaware 19808
(302) 996–3331
(302) 996–5818 (fax)

Conventions Used in This Manual

Italic typewriter font is used for:

- Arguments to macros or functions
- References to chapter, appendix, and book titles

Boldface is used for:

- Chapter, section, and list heading
- Warnings and notes

`Typewriter` font is used for:

- Program examples
- Error messages and computer responses
- Anything typed in verbatim

1 Features and Compatibility

1.1 Overview

The **Eagle Network Security Management System**^{tm} is a suite of software products that resides between your local-area-network (LAN) and another network. The other network can be the outside world (i.e., the Internet) or another of your own LAN's, local or remote.

The **Eagle** is designed to isolate one network from another, allowing only those connections and services that you permitted. The **Eagle** can detect suspicious activity (as you define it) and alert you through a fax, pager, electronic mail or audible alarm. The **Eaglet Subnet Partitioner**^{tm} is designed to work with the **Eagle** to compartmentalize important intranet workgroups or sub-LANS, offering them the same protection that **Eagle** provides.

Just having one (or several) **Eagles** *on your network does not obviate the need for having a coherent, well thought out site security plan. We cannot stress too strongly the importance of doing this planning before there is a catastrophe. Several documents that can assist in this planning are listed in the preface to this manual.*

Figure **??** illustrates the functional components of the **Eagle**, and their relationships.

The **Eagle** features include:

- dual communication interface.
  IP packets are reencapsulated to hide local network topology.
  isolation reduces risk from viruses, trojan horses, and worms.
  maintains the confidentiality of your network.
- host authorization both into and out of your network, as well as within it
  you choose which hosts are allowed.

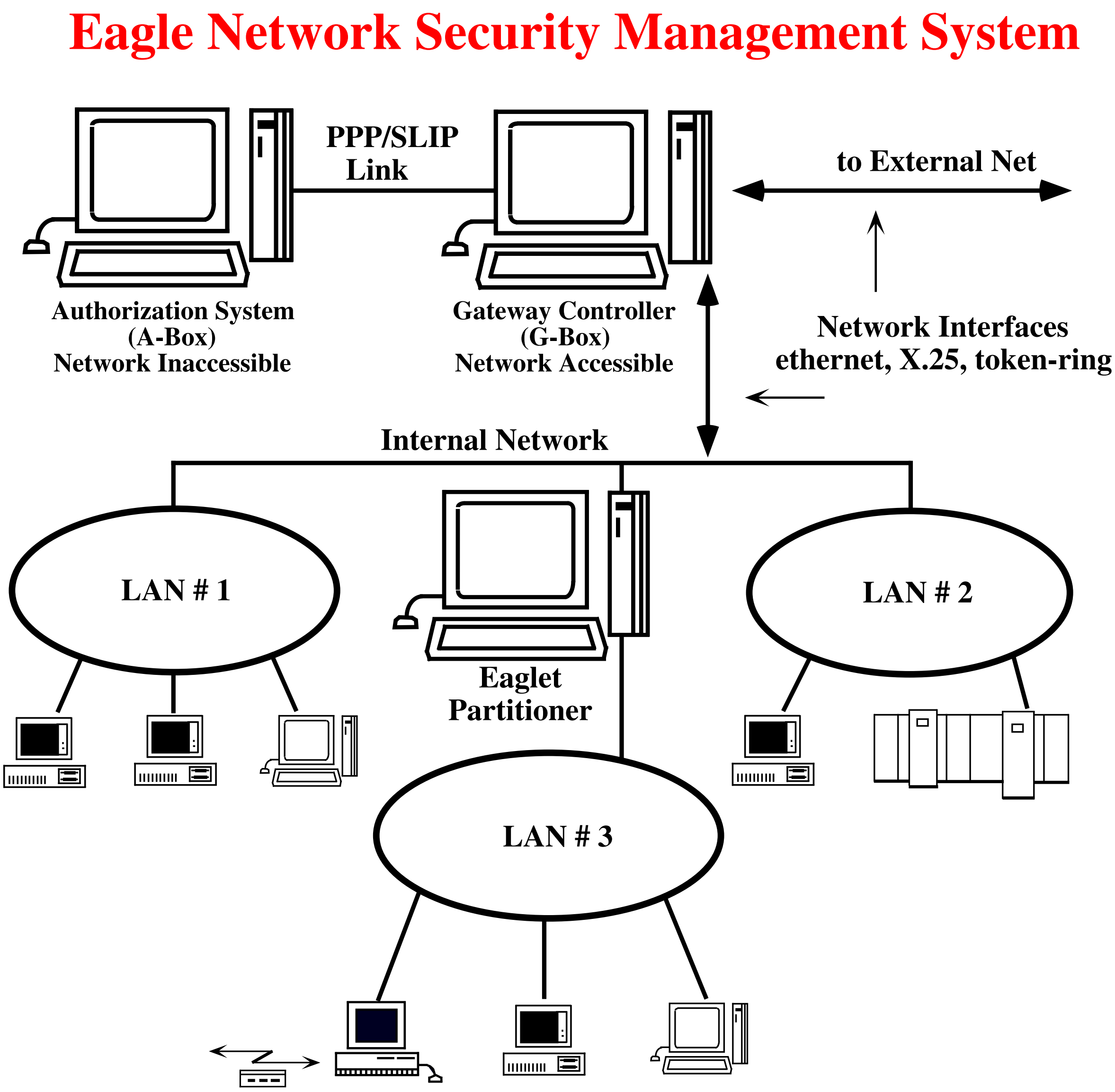

Figure 1.1: Overview of the **Eagle**.

you choose when access is permitted.

you choose what types of communication are allowed.

- monitoring and tracing of suspicious activity

  dynamic activity monitor.

  you establish what constitutes suspicious activity.

  notification of network manager when connection attempts exceed limits.

  traceroute capability to identify offender.

- immediate notification of suspicious activity via:

  display

  audio alarm

  electronic mail

  pager

  fax

  client program of your choice

- service processing

  remote terminal access (`telnet`) modified for enhanced security.

  file transfer (`ftp`) `put` and `get` service selectivity.

  generic packet passer for miscellaneous services you choose to provide.

- user authentication

  identifies who is initiating network traffic.

  restrict `telnet` and `ftp` to particular users or groups.

  optionally authenticates using smart-card technology.

- audit log

  daily report of all activity.

  automatic archiving of previous day's activities.

  easy conversion of log file to fixed-width database formats.

- hardware host authentication via **EtherGuard**

  workstation and PC Ethernet-to-IP address verification.

  detection and shutdown of remote dialup connection through a PC.

- transparency and ease of integration

  uses standard communication port numbers.

  compatible with X.25, Token-Ring, Ethernet, and FDDI.

  no new user commands must be learned.

- system self protection

  no remote login to the **Eagle**; login from console only.

  no user accounts on the **Eagle**.

  single authorization file maintained only by the **Eagle**.

  file changes restricted to authorization machine console only.

  file cannot be remotely accessed.

  file is rule-based and can be customized by the user.

  automatic shutdown of gateway if the program code is altered.

  non-root processes are not allowed to run on gateway.

- subnet isolation with **Eaglet** partitioner

  connection authorization, monitoring, and tracing across subnets.

  one source (**Eagle**) for authorization file and rules.

1.2 Introduction

Welcome to our user's guide for the family of *Raptor* products. *Raptor* designed its products as a response to customers' need for far greater network security. Since security was not an issue when *TCP/IP* was designed, there are many security holes in the *TCP/IP* suite of communication protocols. There are also many ways that a user can achieve privileged status and bypass system-mandated security policies.

Since the **Eagle** (and its associated product, the **Eaglet**) limit the allowed network services to our modified versions of `ftp` (file transfer) and `telnet` (remote login) daemons, and since all network packet headers are rewritten to enforce security for the local network, expectation theory can demonstrate mathematically provable levels of security. The **Eagle** family greatly simplifies the number of 'system dependent' procedures and checklists that your administrator must follow, in essence providing maximum security with minimum inconvenience.

An **Eagle** consists of two discrete processors - the gateway (or **G**) box and the authorization (or **A**) box. The two systems are connected by a private, non-network link. The link is not network accessible, even to privileged users on the gateway box. The **A** box can thus be physically secured. All connections must be approved by the **G** box per rules of access contained in an authorization rule database that resides on the **A** box. There is no way the **A** box's rules can be accessed from the network. The **G** box has read-only access, but only via the private link between the two.

Normal users *never* log onto either system; it is used only as a transit relay. As such, there is no opportunity for them to 'break' the system. Should someone ever determine a way around this, a second level of security is incorporated in the **Eagle**

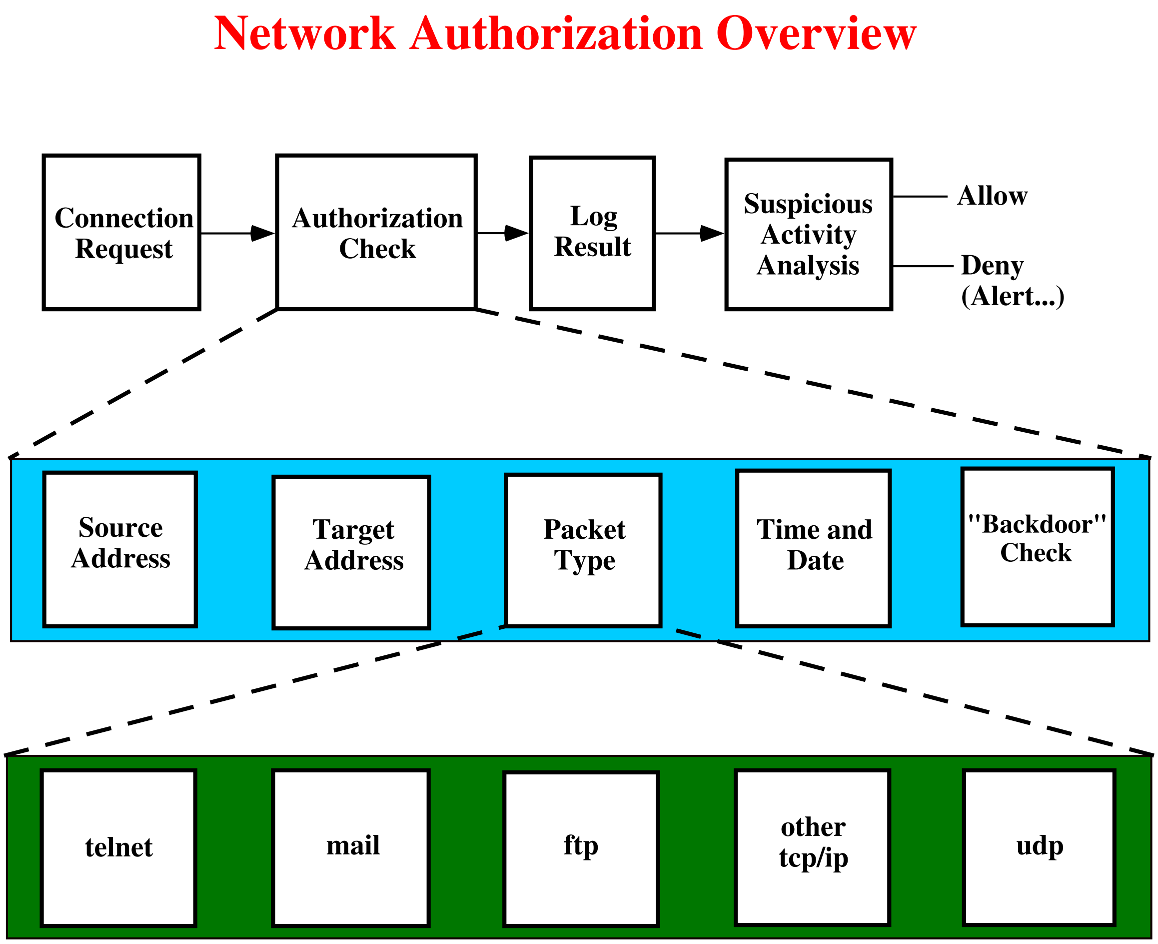

Figure 1.2: Network Authorization Overview of the **Eagle**.

which constantly examines the integrity of the **A** and the **G** box. If it finds active user logins or other processes it does not recognize as legitimate, it will kill them to assure security. If the **Eagle** detects that it has been tampered with, it shuts itself down, preventing all remote network access until a verifiably correct version has been reloaded.

The operational phases of the **Eagle** are illustrated in figure **??**.

1.3 Major Features

In the next few sections we briefly introduce the major features of the **Eagle Network Security Management System**. For further details about each of the features, please refer to the appropriate chapter in this manual.

1.3.1 Network Isolation

The **Eagle** provides security by completely isolating all external networks from your internal network. Except for e-mail via the Simple Mail Transport Protocol (SMTP), no *IP* data from an outside host can directly reach a host on the inside unless you permit it. The **Eagle** lets you specify, for each machine on your network, exactly what kind of communication will be allowed and when. For example, you can authorize physically remote users to upload data to your system during working hours and yet allow no access of any kind during off-hours. Similarly, you can restrict your users' access to outside hosts.

The **Eagle** repackages all *IP* packets it passes, rewriting packet headers to hide information about your internal network. This decreases your vulnerability, as potential attackers are given no information to substantiate or expand on what they may have gleaned from other sources.

1.3.2 Authorization Rule Base

There is a powerful, centrally controlled rule-based system (set up and tuned by the system administrator) which controls access to your network. The rules determine which hosts are authorized to get into or out of the protected network, when access is permitted, which types of access are permitted, and what frequency of successful or unsuccessful connections to the protected network is cause for alarm. Unless a specific rule permits access into or out of your protected network, access is denied.

The authorizations can only be added or modified by the Security Administrator physically sitting at the system console. The rule base cannot be modified remotely, and can only be viewed from the **G Box** console.

1.3.3 Dynamic Activity Monitoring

You can define, and the `Eagle` can monitor, what is considered 'suspicious' or unusual activity. For example, if a node on your network normally connects to a server 4-5 times a day and suddenly makes 50 connections in a single morning, this is suspicious, even if the users and the machine are fully authorized to make these connections. All connections - those allowed and those denied by the gateway - are logged and can be fed directly into any of a number of database systems for processing and report generation.

Depending on the frequency and the magnitude of the suspicious acts, alerts are issued via electronic mail, pager, or fax - directly to appropriate authorities as incidents are taking place, rather than 'after the fact'.

1.3.4 Service Daemons

As installed, the **Eagle** software blocks the passage of all network services into and out of your network, with the exception of SMTP (electronic mail). Since e-mail is a universal service customers want, it passes transparently and you need do nothing to the Eagle software to configure it. You can enable other network services to pass under the control of a powerful rules-based authorization file. Since most customers want to be able to use the common ftp and telnet services, the ability to permit the passage of these services and to control who can use them, both inside of and outside of your Eagle-secured network. is a standard part of the software Further, you can choose to permit the passage of other network services, such as NNTP, NTP, etc.) on a service-by-service basis–and limit the use of them on a user-by-user basis–with our flexible Generic Service Passer.

1.3.5 User Authentication

The **Eagle** supplements any user-authentication technologies that a network or system may employ such as Kerberos or Secure ID^{tm} from Security Dynamics Incorporated. If you consider user authentication to be similar to the PIN number used on a telephone company credit card, the **Eagle** may be considered a device which controls the phones from which your card can be used, when they can be used, and what kinds of calls can be made–and you still have your card and PIN.

The **Eagle** is primarily a network/host authenticator. That is, network traffic thru the gateway is restricted to a particular host or set of hosts identified by network name or address. **Eagle** can also identify *who* is initiating the network traffic. These users must identify themselves with a user name and a password or by SecurID unique key and passcode. This authentication is *in addition* to any other form of authorization/authentication required by the node being accessed or by the application being executed.

1.3.6 Message Log

A log is maintained of all messages, describing the opening, closing and denial (where appropriate) of all connections. **Eagle** system errors are also sent to the log, as is diagnostic information if suspicious activity is detected. This represents an audit trail of the activity on your gateway. It provides valuable clues in the event of an attempted security breach on your network. In addition to saving log messages to a permanent area on disk, `Display` provides a visual representation of the gateway's current state on the system console.

1.3.7 Etherguard

The **Etherguard** can assure that a local connecting host is not 'spoofing' its IP address by providing authentication of its hardware Ethernet address with the IP address it broadcasts, thus preventing a user from masquerading as a trusted machine.

The **Eagle** can close the very dangerous 'back-door' left open by individuals who have IBM compatible or Macintosh^{tm} PC's on their desks with dial-in modems attached. Access to your nework can be restricted on such PC's so that dial-in users are disallowed access to the rest of the network, or to the outside network. This is done via **Eagle**-specific versions of telnet software for the PC's which check for use of the modem and disable communication as dictated by site security policies and the **Eagle** authorization rules.

1.3.8 Vulture

Vulture detects processes running on the **G** box that are (a) not part of the underlying operating system and (b) not part of the **Eagle** software, and kills them.

1.3.9 Secure Checksum

At installation time, the **Eagle** computes and stores a secure checksum for itself. This checksum is periodically compared with the installation benchmark during normal **Eagle** operations. If any change is detected, the **Eagle** shuts itself down (making your network inaccessible) and notifies system administration.

1.3.10 Eaglet

An **Eaglet** performs functions similar to those of the **Eagle**. It lets you partition your overall network into isolated subnets. These subnets are, by default, isolated from the the outside network and from the rest of your internal network unless you specifically authorize access.

1.4 Comparison with Routers

Filtering routers are often proposed as adequate 'firewall' (network security) devices; however, they have inherent weaknesses which make them more vulnerable than the comprehensive, dedicated **Eagle**. Routers are unable to detect suspicious activity; to communicate via fax or pager, or to discriminate between modes of file transfer. They have no 'failsafe' mechanism if someone breaks through, and they have limited speed, since the vendor - not the customer - chooses the cpu. All these features are included in the **Eagle**. Secondly, routers have limited logging capability which makes it difficult to detect trends in system use which may indicate suspicious activity. The **Eagle** has comprehensive logging capabilities. Finally, since they are diskless, routers store control information in their own local memory. Permission changes must either be done manually (a practice which is tedious and error-prone) or by remote access (which can be exploited as a network security hole). The **Eagle** suffers none of these shortcomings. We designed it from the outset to be unbreachable from the outside world.

There are several local area network security devices on the market today. Most require smart card technology, encryption or both. There are few wide area network security devices available, and all of them require that you own, maintain, and control the physical media you use for communication. If you use a common carrier such as a telephone company leased line for your inter-site communication, you are vulnerable to other parties using the carrier service. The **Eagle** uses a consistent, rational approach that permits secure communication across 'hostile' territory regardless of physical transport.

1.5 Hardware Platforms

Raptor's software products require two SPARC compatible workstations running SUN-OS Version 4.1 or greater, two RS/6000 workstations running AIX Version 3.2 or greater, or two Hewlett-Packard 9000/7xx workstations. You can combine any two of these hardware platforms to create your **Eagle** system if you wish, but this requires a customized installation not supported by the standard **Eagle** installation software. The first of the two workstations serves as the **G Box**, and the second as the **A Box**. The purpose of these machines is described in Chapter **??**.

The Gateway computer must have dual ethernet cards, a minimum or 400 megabytes of disk space, and 32 megabytes of memory. We do not recommend that customers use low-end equipment for the Gateway computer, particularly if your network is large. The Authorization computer must have a minimum of 400 megabytes of disk space, and 16 megabytes of memory. Although the Authorization computer need not be as powerful and as fully equipped as the Gateway computer, bf Raptor Systems recommends that the Authorization computer be the same model as the Gateway box, and that it be equipped with the same hardware, in order that you may be able to use it as a substutute Gateway computer in the event of hardware malfunction. A serial cable with a null modem is required to setup a private link between the Gateway to the Authorization computer.

2 Components of the Eagle

2.1 Overview

The **Eagle** is designed to isolate one network from another, permitting only those connections and services you specify. Typical applications for this gateway would be to limit access to a secure network from the outside, or to screen connections between an exceptionally busy network and the Internet. As installed, the Eagle software blocks the passage of all network services into and out of your network, with the exception of SMTP (electronic mail). Since e-mail is a universal service customers want, it passes transparently and you need do nothing to the Eagle software to configure it. You can enable other network services to pass under the control of a powerful rules-based authorization file. Since most customers want to be able to use the common ftp and telnet services, the ability to permit the passage of these services and to control who can use them, both inside of and outside of your Eagle-secured network. is a standard part of the software Further, you can choose to permit the passage of other network services on a service-by-service basis–and limit the use of them on a user-by-user basis–with our flexible Generic Service Passer.

The **Eagle** can also be used to act as a 'gatekeeper' in a network with multiple levels of security, to ensure that data is not accidentally transferred from a highly secure machine to a less secure machine. As we have seen in the preceding chapter, the **Eagle** can watch the types of connections being completed or refused, then alert systems management if there is any unusual activity. This is called **Dynamic Activity Monitoring**. (The details of specifying suitable criteria for this is described in Section **??**).

Additional gateway servers called **Eaglets** may be installed if you wish to partition a network. **Eaglets** share the authorization function of the **Eagle**, but control traffic among sub-networks. Since an **Eaglet** uses the same authorization file as the **Eagle**, only one copy of the authorization file is necessary, thereby significantly easing system administration requirements and eliminating authorization skew.

In virtually all cases hereafter, the capabilities and functions ascribed to the **Eagle** also apply to **Eaglets**.

Raptor's software products protect networks that pass *IP* packets only. They completely block all other network protocols.

2.2 Hardware Components

The **Eagle** requires two high-performance workstations connected via a PPP (Point-to-Point Protocol) or SLIP (Serial Line Interface Protocol) connection. One workstation, the **G** box, has two separate network interfaces and serves as a gateway between the outside world and your network. The other workstation, the **A** box, is only connected to your network via a private PPP or SLIP link to the **G** box. The **A** box holds the **Eagle's** authorization rules and should be locked in a secure area to prevent tampering. No one can access the **A** box from a network connection, even from the **G** box. The **Eagle** system administrator may only access the **A** box from its console.

If you choose to partition your network into secure isolated subnets, an additional workstation is required for each **Eaglet**.

Finally, you will need a modem if you want to be alerted to 'suspicious activity' via a fax or pager.

2.2.1 Authorization Machine

The **A** box stores the authorizations for machines which are allowed to use your network (i.e., external machines you have decided to allow into your network, and internal machines you permit to leave your network or its subnets). You are able to authorize a machine by *hostname* or *IP address*, by *service* (i.e, `ftp`, `telnet`, `ntp`, etc.), and by *time* (i.e., Monday thru Friday 8AM to 4:30PM). Combining these criteria allows tremendous flexibility as you secure your network. You may further restrict `ftp` to *getting* (downloading) or *putting* (uploading) files. The **A** box is connected to the **G** box through a private network link, physically separate from the protected network. The Security Administrator can only change authorizations by sitting at the **A** box console. The rule base cannot be viewed or modified remotely, and the **A** box cannot be used for any purpose other than authorization.

2.2.2 Gateway Machine

The **G** box acts as a gateway with two network interfaces (Ethernet or any other transport that passes *IP* packets, such as Token Ring, FDDI, serial line, or a combination). The **G** box accepts connection packets, examines them, and communicates with the **A** box via the private link to check the authorization rules. If the **A** box allows the connection, a link is established but outgoing packets from your network are reassembled to hide information about your network. Each packet looks as if it came from the **G** box although it may have come from anywhere within your LAN. Thus packets leaving your network contain information only about the **G** box and no other information about the internal structure of your network.

The **G** box must be dedicated to running the **Eagle** software and cannot be used for other purposes. The **Vulture** subcomponent of the **Eagle** (described in section **??**) ensures that no other processes can be run.

The **G** box is connected to the **A** box with a null-modem RS-232 cable and communicates via PPP or SLIP. This link is not reachable from either the outside network or from other hosts on your local network.

2.2.3 Eaglet Subnet Partitioner

The hardware for **Eaglet** subnet partitioner is a workstation similar to the **G** box with two network interfaces running the same gateway control software. The **A** box communicates with an **Eaglet** in an encrypted protocol. There is no disk copy of the authorization file on an **Eaglet**. Like the **G** box, an **Eaglet** repackages *IP* packets to hide the implementation details of the subnet from any potentially malicious eavesdropping.

The **A** box can control any number of **Eaglets**.

2.2.4 Fax/Pager Modem

You may attach an external Class 2 fax modem to the **G** box to send fax and/or pager notification of suspicious activity. See Chapter **??** for a discussion of suspicious activity and notification. An example of the fax output is shown in Appendix **??**.

If you want to send messages to your pager and not to a fax, you can use any V32.bis modem. The system performs pager notification by dialing your pager number via modem and leaving a numeric code on your pager. If your pager has an alphanumeric display, **Eagle** messages (refer to Appendix **??**) are sent when appropriate.

To prevent outside users from attempting to log onto your **G** box, configure your modem to refuse incoming connections.

2.3 Software Components

The **Eagle Network Security Management System** consists of several software components. The gateway control program (`gwcontrol`) is the heart of the system. It handles network connection requests from remote systems and from systems on your local network. After consulting its configuration (or authorization) file, `gwcontrol` either completes the connection or denies access. `Gwcontrol` maintains a list of active connections. For added security, we have provided modified versions of the standard service daemons `ftpd` and `telnetd` which interact with `gwcontrol` to process service requests. The `display` program shows the state of the gateway and current connections based on `gwcontrol`'s information. An auxiliary program, the **Vulture**, kills programs which aren't allowed to run on the **G** box. The following sections describe these programs in detail.

While the **Eagle** routes packets entering your network, it is more capable than a standard router: it disassembles incoming packets and constructs new ones before sending them out to the destination host. This guarantees that there are no 'cracks' for undefined packet types to exploit.

The **Eagle** maintains a complete transaction log. It alerts system administration whenever suspicious activity (e.g., four failed connect attempts from the same site within five minutes) is detected. As soon as the **Eagle** determines that something potentially harmful is happening, it logs the source of the offending packets and the route they travel. Recall that you, the system administrator, define what constitutes 'suspicious activity'.

If you have a modem attached to your system, the **Eagle** can notify appointed individuals via pager or fax whenever intrusions are detected. The **Eagle** can also send alerts to a pager or to the system's loudspeaker; via electronic mail; or to a client program of your choice for processing. Refer to Chapter **??** for more details.

The **Eagle** software is auto-encrypted to discourage attempts to reverse-engineer it. It comes with a secure hash function checksum so that system administrators can routinely confirm that no one has tampered with the code. If the **Eagle** has been altered, it shuts itself down, preventing all remote network access until a verifiably correct version is reloaded. To provide assurance that there is no 'back door' which might be exploited, the source code can be certified by an external agency using a secure checksum.

2.3.1 Gateway Control

As mentioned previously, `gwcontrol` is the heart of the **Eagle**. On startup, it reads the authorization file, securely stored on the **A Box** and waits for connection or service request. The authorization file is a powerful, centrally controlled rule-based system (set up and tuned by the system administrator) to control access to your network. The rules determine which hosts are authorized to get into or out of the protected network, when access through the protected network is permitted, which types of access are permitted, and what frequency of successful or unsuccessful access to the protected network is cause for alarm. Unless a specific rule permits access into or out of your protected network, access is denied. See Chapter **??** for more details.

When a `telnet` or `ftp` connection is made, the modified daemon contacts `gwcontrol` with the source and destination machine names. `Gwcontrol` then looks through the list of allowed connections, and responds either with a completed connection or denial of access accordingly. If it allows a connection, it then adds that to its list of current connections. The following information is retained for each connection and is visually presented using the `display` program on the **G** box's console:

Attribute	Description
Service	Type of service (`telnet` or `ftp`)
Remote host	The name (if not known, *IP* address) of the source machine
Local host	The name or *IP* address of the destination machine
ID	The process id of the connection's daemon
Connection duration	The time that the connection was established

`Gwcontrol` maintains a log of all attempted and completed connections (including when initiated and terminated). System errors will also be sent to the log (see Appendix **??**), as will any diagnostic information if suspicious activity is detected. Archived log files are created on a daily basis. This makes it easier to track down possible security violations by limiting the log entries that have to be examined.

The system administrator can immediately terminate any connection with `display`. The process id of the connection's daemon is sent a 'terminate' signal and a message is written in the log. You must be physically at the **G** box's console to terminate a connection.

If a possible security breach is detected, a trace of the network route back to the offending host will be written into the logfile.

The log file is kept on the **G** box. For security reasons, there is no way that you can reach this machine from inside your network. To specify where you are going to archive this file, you must edit the `changelogfile` script to mail it to the desired system (see Section **??**).

2.3.2 Dynamic Activity Monitoring

The **Eagle** is able to detect what you deem to be 'suspicious' activity. What is suspicious for one site may be acceptable for another, so the rules must reflect your site's unique requirements. While some rules, such as *never allow data to be transferred from a machine of high security to one of lower security* are clear, others may not be. For instance, the fact that a particular rule is triggered once may mean nothing except a mistake by a user typing his/her password. If, however, a rule is triggered many times in a brief period, it may indicate that something suspicious is being attempted (or worse, is actually happening).

To define what you deem 'suspicious', you can specify a set of sliding time windows together with associated 'thresholds' for connection attempts. The **Eagle** keeps track of which rules have been invoked either to permit or to deny connections during each window (the current time is always the end of each window). You specify how often during these windows it is acceptable for a rule to be active. For example, if your site has been averaging four or five connections from the *mit.edu* network per day, having six or seven in a given day is probably not a notable incident. However, if there have been ten or more in a single day, you may want to know about it. If there have been more than thirty, you will certainly want to be alerted, especially if it is in the middle of the night (hence the pager alert system in the **Eagle**). Each of these time windows and thresholds can be separately configured for each access rule.

The **Eagle** comes with default thresholds for all time windows. Over-riding these defaults is simply a matter of changing the authorization rule database.

Suspicious activity falls into several categories or severity levels. These are:

Severity	**Description**
`Emergency`	`gwcontrol` has failed and the system will no longer allow traffic to flow in or out of the **G** box.
`Critical`	security is still in place, however an ancillary program has failed.
`Alert`	a security rule has been triggered.
`Error`	normal gateway operation cannot complete successfully.
`Warning`	occurs when there is a recoverable error.
`Notice`	occurs when someone attempts to reach a machine and is denied.
`Info`	standard operational messages which indicate the gateway is operating properly.

All 'suspicious' activity is recorded in `gwcontrol's` log file. You can be notified of 'suspicious' activity by audio alarm, electronic mail, pager, fax, or client program. See Chapter **??** for a full description of notification.

Figure **??** illustrates how the **Dynamic Activity Monitor** helps to protect your network.

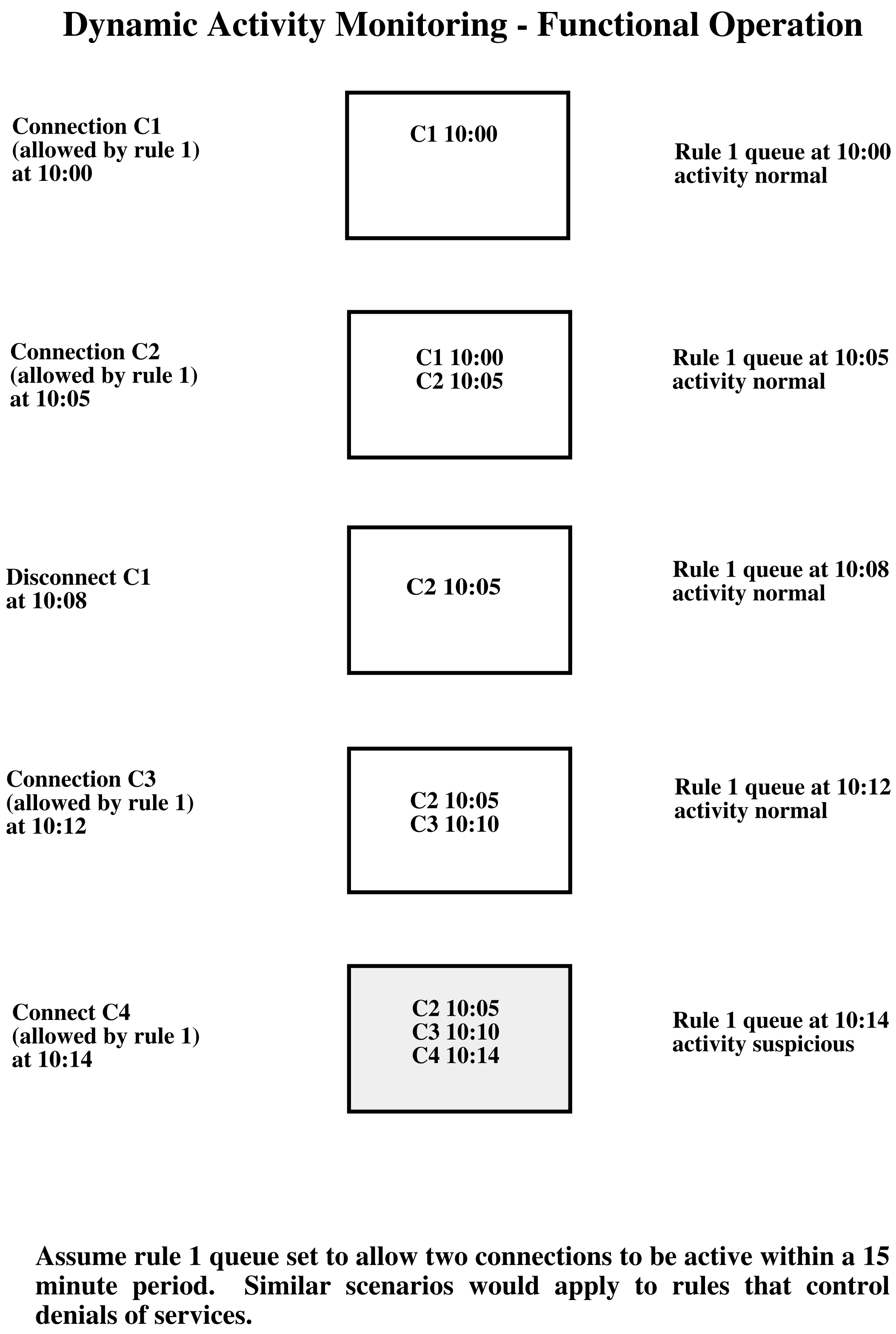

Figure 2.1: Overview of the **Dynamic Activity Monitor**.

2.3.3 Display

`Display` provides a visual representation of the gateway's current state. It also allows you to reload the **Eagle** authorization rules file and terminate active connections, but it is not required for the gateway to operate correctly.

`Display` uses the `curses` terminal control library for all its screen operations so it can run on any terminal or terminal emulator, including an X-Windows `xterm`. Thus the **Eagle** may be used without a bit-mapped console display; any terminal may be plugged into an RS-232 port on the **G** box.

On startup, `display` connects to `gwcontrol`. It displays current connections in the top half of the screen and the last few log file entries in the bottom half. The display is updated every four seconds. The top line contains the current time and the current number of active connections. If there are more connections than will fit in the top portion of the screen, `display` will show subsets of the connection list at four-second intervals.

`Display` also accepts commands from the terminal. They are:

Command	Description
`Refresh`	erases the screen and redraws it. This is useful if the screen becomes unreadable.
`Quit`	exits `display`.
`Kill`	closes a currently active connection based on ID (the process id of the connection's daemon.
`Shutdown`	closes all active connections and stops `gwcontrol`.
`Startup`	starts `gwcontrol` and enables the gateway.
`Configure`	forces `gwcontrol` to re-read its configuration file (use this command after each change to the authorization rule set).
`Print log`	prints the current log to the default printer.

If you are running `display` on the system console, normal console messages are captured and displayed along with **Eagle** messages. These messages are not entered into the **Eagle's** log file. They appear as:

`CONSOLE:` *console message*

Selecting `refresh` will clear the display of console messages. If you are running `display` in an X Windows environment, you can direct console messages to a console window.

The installation script creates an account with username `display`. When you log in to this account, the `display` program starts automatically. `display` runs with superuser authority. When you exit `display`, you will then be logged out. You may also also access `display` after logging in as `root` by typing `su display`.

2.3.4 Service Daemons

As installed, the Eagle software blocks the passage of all network services into and out of your network, with the exception of SMTP (electronic mail). You can enable other network services to pass under the control of a powerful rules-based authorization file. Further, you can choose to permit the passage of other network services on a service-by-service basis–and limit the use of them on a user-by-user basis–with our flexible Generic Service Passer. Although other network services can be run on the **G** box, this practice is discouraged since it could expose you to security risks that may exist in those services.

TELNETD and FTPD

Next, to electronic mail, the most widely used Internet services are **telnet** (remote login) and **ftp** (file transfer). The **Eagle** software has built-in, secure support for **telnet** and **ftp**. In addition, you can control the **Eagle** versions of these facilities to limit their use according to your security policy.

The **Eagle** software installation replaces the standard **telnet** and **ftp** "service daemons" (`telnetd` and `ftpd`, respectively) on your Gateway host with **Raptor Systems'** modified, secure versions. The user interface for **telnet** and **ftp** in an **Eagle**-secured network differs slightly from the standard interface. **Raptor Systems** describes its **telnet** and **ftp** facilities as *passthrough* facilities. That is, users inside the network interact with the **Eagle** `telnetd` and `ftpd` service daemons on the Gateway host to access remote systems. Once connections between the internal and external hosts are authenticated, data between them passes through the Gateway machine under the control of the secure `telnetd` and `ftpd` service daemons.

The **Eagle** offers facilities for *limiting telnet and ftp services*. To begin with, connections between internal and external hosts must be authorized in the **Eagle** authorization file (see Chapter **??**). Second, individual users may optionally be required to identify themselves, with an encrypted password, before being permitted to use **ftp** or `telnet` to access external hosts. Finally, with respect to `ftp`, a subset of this service daemon's commands may be specified for each host in the authorization file. There are two such limits for `ftpd`: `putonly` and `getonly`. They allow the system administrator to separately permit write-only or read-only access. `Ftpd` screens incoming commands and returns an error message if a command is disallowed.

The **Eagle** may require up to six pieces of information to determine if a `ftp` or `telnet` session is authorized. The information needed is

1. destination user name (the user name on the remote system)

2. destination address

3. destination password (the password on the remote system)
4. gateway user name (the user name on the local system)
5. gateway password (the password on the local system)
6. alternate TCP/IP port number or service name

The destination user name and address is always required. Before `gwcontrol` can determine what additional user authentication information may be needed, it must have the destination address to determine which authorization rule applies.

The gateway user name and password are required whenever user authentication is enabled. Refer to Chapter **??** for more information about user authentication.

The alternate TCP/IP port number is used to access (via `telnet` a service available on the specified port number on a remote host.

In the example below we assume that the following line is in the **Eagle's** configuration file (see Chapter **??**):

```
faraway.outbound.com allow mymachine
```

This permits users on `faraway.outbound.com` (a system outside the **Eagle**-secured network) to connect to `mymachine` (inside the secure network) via *passthrough* `telnet` or `ftp`. (In this example, `firewall.xxx.com` is the name of the **Eagle**.) Figure **??** illustrates a `telnet` connection from `faraway` to `mymachine`:

Note the differences between this example and the standard `telnet` connection dialogue.

1. The user specifies the **Eagle** Gateway as the destination host, not the true destination host.
2. The **Eagle** Gateway prompts for the true destination hostname
3. The destination host prompts for a username and password before permitting the login.

Although this example does not illustrate **user authentication**, had it been required, the **Eagle** Gateway would have prompted for a username and password and verified them before connecting to the internal host.

Having been authorized to connect by the **Eagle** Gateway, the user is now logged in as user `username` on `mymachine` via *passthrough* `telnet`. The **Eagle** will continue to transparently pass characters between the two systems until the user logs out.

faraway% `telnet firewall.xxx.com`	*Connect via the* **Eagle** *for passthrough*
Trying 192.92.2.1...	*IP address of the* **Eagle**.
Connected to firewall.xxx.com.	
Escape character is ']. '	
This is a private network. Enter hostname.	*Message in the motd file.*
Hostname: **mymachine**	
	Verifying connection.
SunOS UNIX (mymachine)	*Host banner message.*
login: **username**	*Name to login as.*
Password:	*Password not echoed to screen.*

Figure 2.2: Sample TELNET Dialog

.

As you would expect, the logon dialogue for `ftp` also differs from the standard ftp version.

Assume that the **Eagle's** configuration file contains the following line:

```
faraway.outbound.com allow mymachine(ftp.getonly)
```

This allows users on `faraway.outbound.com` (a remote system outside the **Eagle**-secured network) access to the internal host `mymachine` via *passthrough* `ftp`, but limits them to downloading files. As with the previous example, the name of the **Eagle** is `firewall.xxx.com`. Thus, users on `faraway` who want to connect to `mymachine` see the dialog illustrated in figure **??**.

Note the differences between this example and the standard `ftp` connection dialogue.

1. The user specifies the **Eagle** Gateway as the destination host, not the true destination host.
2. The **Eagle** Gateway prompts for the true destination hostname and username, in the format **username@hostname**
3. The **Eagle** Gateway prompts for the user's password on the destination host.
4. The user is allowed to "get" (download) a file, but is denied permission to "put" (upload) a file.

faraway% `ftp firewall.xxx.com`	*Connect via the* **Eagle** *for passthrough*
Connected to firewall.xxx.com.	
220 firewall FTP server ready.	
Name: **username@mymachine**	*User and machine to access.*
331 Password required for username.	
Password:	*Password not echoed to screen.*
230 User username logged in.	
ftp: **get afile**	*Allowed to get (download) files.*
150 ASCII data connection for afile (9999 bytes).	
226 ASCII Transfer complete.	
9999 bytes received in 0.10 seconds (99 Kbytes/sec)	
ftp: **put afile**	*Not allowed to put (upload) files.*
502 Access to this command denied.	
ftp:	

Figure 2.3: Sample FTP Dialog

.

When the connection to the destination host is closed, the service daemons will reconnect to `gwcontrol` to report that the connection is no longer there. `Gwcontrol` then removes that connection from the active list and updates the log.

User authentication provides more flexible control of who accesses hosts in your **Eagle**-protected networks by allowing you to require additional username and password authentication before users can go into or out of your protected network(s). Refer to Chapter **??** for example user authenticated `telnet` and `ftp` dialogs.

The `telnet` and `ftp` daemons display a *message of the day* upon initial contact. You can change this message. Edit `/usr/adm/sg/gateway_motd` to change the `telnet` message, or `/usr/adm/sg/ftp_motd` to change the `ftp` message of the day.

Generic Service Passer

As installed, the Eagle software blocks the passage of all network services into and out of your network, with the exception of SMTP (electronic mail). Since e-mail is a universal service customers want, it passes transparently and you need do nothing to the Eagle software to configure it. You can enable other network services to pass under the control of a powerful rules-based authorization file. Since most customers want to be able to use the common ftp and telnet services, the ability to permit the passage of these services and to control who can use them, both inside of and outside of your Eagle-secured network. is a standard part of the software Further, you can choose to permit the passage of other network services on a service-by-service basis–and limit the use of them on a user-by-user basis–with our flexible Generic Service Passer. The

Generic Service Passer allows you to do so by importing (or exporting) a service from a computer on one side of the **Eagle** to a machine on the other side.

The **Generic Service Passer** waits for connections on specified ports, opens connections to specified hosts, and then simply copies packets from one to the other. For UDP, which has no notion of a connection, a pseudo-connection is made that will close down after one minute of inactivity.

Details on the use of this facility are found in Chapter **??**.

2.3.5 Notify

The `notifyd` daemon works closely with the messaging capability of the **Eagle**. `Notifyd` can alert appointed individuals via pager or fax whenever intrusions are detected, even if the **Eagle** is unattended. Alerts can also be sent to a pager or the system's loudspeaker, via electronic mail, or to a client program of your choice for processing.

Details on the use of this facility are found in Chapter **??**.

2.3.6 EtherGuard^{tm}

For added protection, a site within your network may need to verify that an IP address offered to the **Eagle** by a connecting host within the local network is authentic; in other words, protection against IP address 'spoofing' may be desired. Personal computers are inherently insecure and it is easy for users to change the configuration of the network software on them. **EtherGuard** is a component which uses ARP (Address Resolution Protocol) to check the physical Ethernet address (usually of the network adapter card in the workstation) against the Internet (IP) address for hosts specified in the authorization file. The authentication is performed via an encrypted message. An improper answer generates a message that warns of the possible security breach.

Modems on personal computers present yet another security risk to a computer network. A user with a home PC, for example, could dial up a server PC and then use the latter's local TCP/IP network connection to access secured resources. To provide this additional protection, a second component of **EtherGuard** running on the PC is utilized.

Whenever a connection from the PC is attempted, **EtherGuard** checks to see if the modem is in use; if so, the connection is dropped. This guarantees that you must be sitting at the PC's console to use this PC on the network.

This second component of **EtherGuard** is a modified version of NCSA `telnet` software, supplied by *Raptor Systems*, that provides client support for `telnet` and both server and client support for `ftp`. This should be run in place of other network communication programs on each PC you wish to restrict. Other network programs, such

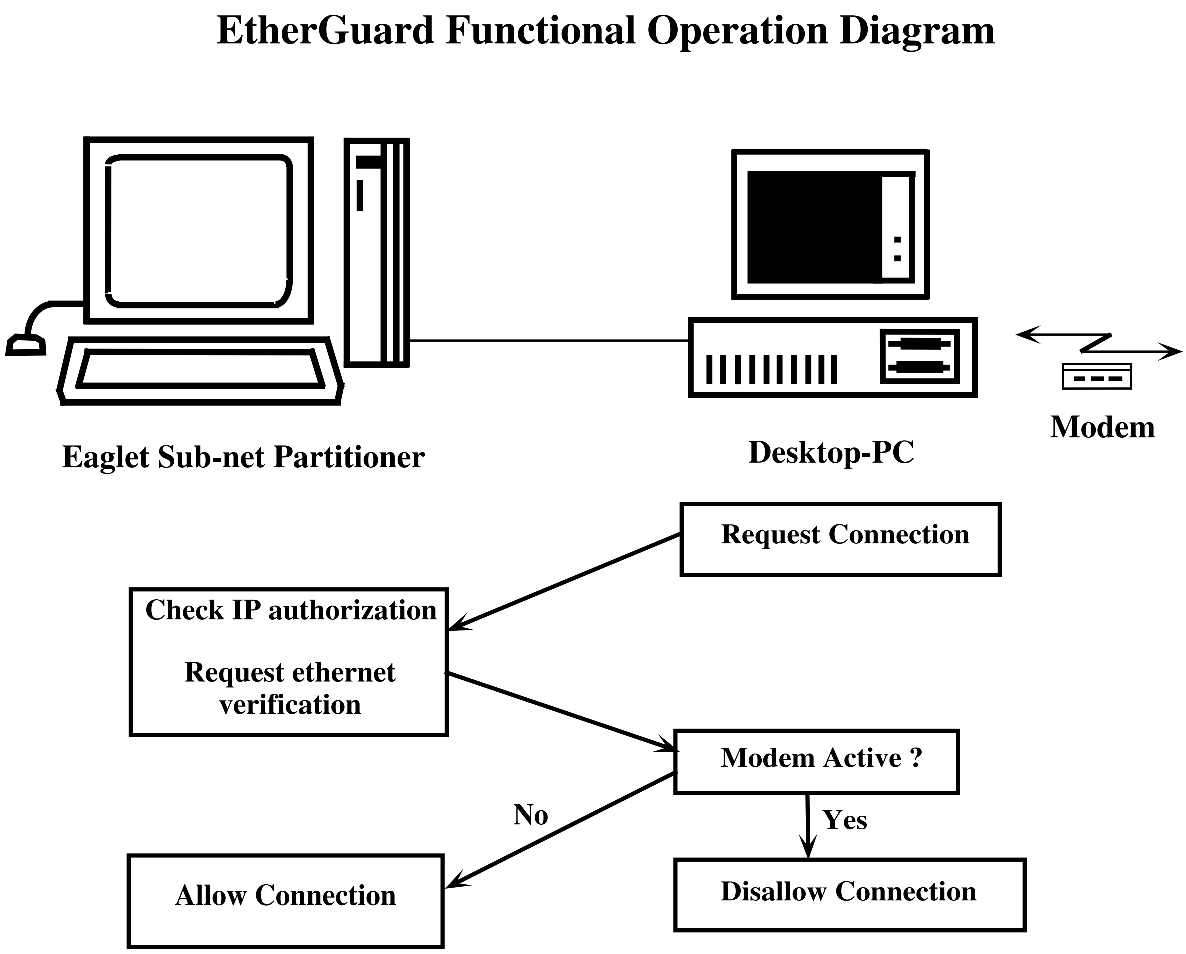

Figure 2.4: Overview of **EtherGuard**.

as PC-NFS, can still be used to give strictly local network access, but remote access is only permitted through NCSA `telnet`.

Figure **??** illustrates the components of **EtherGuard**, and their relationships.

2.3.7 Vulture

The second level of security is handled by **Vulture**. It detects processes running on the **G** box that are (a) not part of the underlying operating system and (b) not part of the **Eagle** software, and kills them. This is also called *active security* to differentiate it from the passive first level of security, which only acts when connection requests are received. By default, the **Vulture's** activation frequency is one minute, but you can change it by creating the file `/usr/adm/sg/vulture.runtime` and placing the new frequency (in seconds) in the file. A value of `-1` disables the **Vulture**. You can exempt non-root administrative processes on an individual basis, such as the `sendmail` program, which under AIX runs under the authority of the `bin` user, from the **Vulture** by specifying a userid in the `/usr/adm/sg/vulture.runtime` file. For example,

    90 bin

This example sets the **Vulture** activation frequency to 90 seconds, and exempts the user "bin".

2.3.8 Secure Checksum

At installation time, the **Eagle** computes and stores a secure checksum for itself. This checksum is periodically compared with the installation benchmark during normal **Eagle** operations. If any change is detected, the **Eagle** shuts itself down (making your network inaccessible) and notifies system administration.

3 Authorization Rule Database

3.1 Overview

The authorization file, `gateway.cf`, on the **A Box** contains access rules for the **Eagle**. The `gwcontrol` program reads this file every time it starts or when the `display` program executes a `configure` command. Thus, it serves as a configuration as well as an authorization file. Each line of this file is a compact and powerful **rule** determining access criteria among various hosts on either side of the **Eagle**. Entries in `gateway.cf` determine the following:

- Which hosts are authorized to get into or out of the protected network.
- When access through the protected network is permitted.
- Which types of access are permitted.
- What frequency of successful or unsuccessful access to the protected network is cause for alarm.

Each rule in the authorization file has the form:

<*thresholds*> *source* `deny` *destination* [*time limits*] (*service limits*)
<*thresholds*> *source* `allow` *destination* [*time limits*] (*service limits*)

Only the **allow/deny** field is required; it must be included for any authorization rule to be recognized. *Italicized* fields are optional and may be omitted. The *source* and *destination* fields allow network access via the **Eagle** to the hosts or networks you specify. The *thresholds*, *time limits*, and *service limits* fields allow you to fine-tune the access you allow. In addition, if you have an **Eaglet** subnet partitioner, you may specify rules which apply only to the **Eaglet**.

The next few sections describe each of the fields in a rule.

3.2 Required Authorization Fields

Each rule must have either the **allow** or **deny** field. If the *source* or *destination* field is blank, then that field matches all machines. Otherwise, the *source* and *destination* fields specify which machines and networks the **Eagle** allows access to and from.

3.2.1 The Default Authorization Rule

The **Eagle** software automatically denies access to a machine if no other rule in the authorization file matches it. This is known as the default rule. You should, however, always put as the first rule in the authorization file an explicit deny:

```
deny
```

This rule reads as *All machines are denied access to all machines.* This single-word statement assures maximum security by disallowing any incoming or outgoing connections. Since this is too restrictive for most applications, succeeding lines will contain rules which allow access to specified hosts and/or networks.

While it is possible for an 'allow-all' rule, similar to the default 'deny-all' rule, to be placed in the authorization file, doing so would defeat the **Eagle's** purpose. Placement of such a rule is therefore strongly discouraged.

3.2.2 Host/Network Designations in Authorization Rules

The *source* and *destination* fields of the authorization file may be very specific, referring to individual hosts, or general, referring to groups of hosts, to networks, or to entire Internet domains, or, if left blank, to all machines everywhere. For purposes of the authorization file, the term 'host' may refer to any of the above and may be either remote (outside your local network) or local (inside your local network). Some examples follow.

host name The name of an individual computer as returned by hostname lookups. This will match only exactly during a verification query by `gwcontrol`. Examples: `export` and `express.army.mil`.

Network name The symbolic name of a network that has been entered into the `/etc/networks` file. The effect is the same as if replaced with the network's internet address. The netmask can be explicitly set using an ampersand (`&`) followed by a mask. The mask can be in 3 different forms: (1) a 'dotted quad'; (2) a hex number; or (3) a decimal number of bits in the host part of the address. If you do not explicitly set the mask in this way, the file `/etc/netmasks` will be searched. If there is no entry for the network, a netmask will be generated based on the class of the network.

Internet address The *IP* address of a machine in 'dotted quad' format. This must be an exact match during verification. A *network* address (fewer than four numbers) will match any machine on a network. Examples: `128.195.28.2` matches the single machine at that address, while `128.57` matches all machines whose addresses start with `128.57` (all hosts on the `128.57` network). The second example may also be written as `128.57.0.0`. For network addresses, a netmask is generated in the same manner as described above in **??**.

Internet domain This is the name of a network or group of networks, or the tail end of a fully-qualified domain name. It matches any fully-qualified domain name whose tail end matches it. This name begins with a '.' to separate it from a possible machine with the same name. Examples : `.edu` and `.cms.xxx.com`. The first matches any machine from the educational (`.edu`) network domain; the second matches all machines on the `cms.xxx.com` commercial domain.

blank If no machine is listed in either the *source* or *destination* field, all machines are matched. Note that the **default** rule (see above) contains blanks in both the *source* and *destination* fields. An address of `0.0.0.0` is the same as leaving the field blank; it matches all machines. Use blank fields with caution.

Under these rules, a single machine can match several different entries in the authorization file. The more specifically a rule matches a particular machine, the greater precedence it will have. If more than one entry matches equally well, then the first match encountered will take precedence. One should apply the following `gwcontrol` rules to disambiguate the sequence of rules in the file:

1. Use an exact match to either a hostname or an numerical internet address.

2. Use the lowest level (or most specific) internet domain or internet address. Therefore, `mine.cms.xxx.com` more closely matches `.cms.xxx.com` than `.xxx.com`. Similarly, `128.195.28.2` more closely matches `128.195.28` than `128.195`.

3. Use either a blank field or `0.0.0.0` for a rule which matches all machines.

3.2.3 Authorization Rules

Authorization rules contain the verb **allow** or **deny**, usually in the following syntax, where *source* and *destination* are hostnames according to the rules stated in section **??**:

source `allow` *destination*
source `deny` *destination*

As noted, *source* refers to the requesting host and *destination* to the requested host. This allows `gwcontrol` to control connections among local machines on your network or between local machines and the Internet, and to control outside access to your local network. Hosts can be specified by using the following types of syntax:

- local name (as known in the subnet), e.g. `alpha`
- fully qualified name (as known to the Internet), e.g. `alpha.subnet.industry.com`
- general domain name (includes all hosts in this domain), specified with the leading '.', e.g. `.industry.com`
- fully qualified *IP* address in 'dotted quad' format, e.g. `192.24.145.1`
- numeric *IP* subnet designation, e.g. `192.24`

For each of the two fields above, *source* and *destination*, multiple hosts can be designated within the same field by separating entries with commas, with or without separating spaces:

```
alpha, .industry.com allow 192.48.145.1, 192.24
```

This rule specifies that local host `alpha` and all hosts in the Internet domain `.industry.com` are permitted to connect to the individual host with IP address `192.48.145.1` and to all hosts on the `192.24` Internet subnet.

Note that specifying more than one host in the *source* field subjects them all to the same time thresholds. A connection from any of the hosts in the list will increment the count. See Section **??** for more information about thresholds.

3.3 Optional Authorization Fields

Authorization rules have these optional fields:

1. thresholds
2. time limits
3. service limits
4. **Eaglet Subnet Partitioner** name

Use *thresholds* to define normal and abnormal activity for your site; see Section **??** for more details. *Time limits* can restrict access to particular times of day or days of the week (Section **??**). *Service limits* can restrict services such as `ftp` or `telnet` for your network (Section **??** for a description of restricting services). Finally, specifying an **Eaglet** name implements rules applied only by **Eagle Subnet Partitioner(s)** in your network (Section **??**).

3.3.1 Dynamic Activity Monitoring

Thresholds specify how many network accesses are considered normal within a defined period of time. A security alert is generated when a threshold is exceeded. For example, users occasionally mis-type their login password. Such an occurrence would not normally be a security concern. However, a large number of password authentication failures within a short time might indicate an attempted security breach. These optional rule specifications, spelling out **how many** accesses are allowed within a **specified time**, permit the **Eagle** to respond to repeated accesses with appropriate alerts. See Chapter **??** to learn how to set up special notices for unusual activity.

The syntax for specifying threshold values for the rules is:

```
< accesses / period, ...  >
```

Note that this field is enclosed in angle brackets '<>'. This syntax calls for a list of entries, separated by commas or blanks, which describe how many accesses to permit in a time period before triggering an Alert. The time period is a numeric value which may be followed by a qualifier (*minutes*, *hours* or *days*). If no qualifier is specified, the default value is *minutes*. There can be any number of time-window values within the *threshold* field. The default thresholds are:

```
< 3/5 mins, 5/15 mins, 10/hour, 25/day, 100/7 days >
```

These defaults allow 3 accesses in a 5 minute interval; 5 in 15 minutes; 10 in an hour; 25 in a day, and 100 in a week.

A shorthand notation allows you to omit the time period specification if only the counts are to be changed from the default. To change the number of accesses allowed in the 1 hour period to 15, without changing the other default subfields, one would enter:

```
< , , 15, , >
```

Notice that not only could the *hour* qualifier be omitted (it is the default for that subfield), but blank fields, separated by commas, specify that the entries for those subfields stay the same. Thus, the above rule is the same as the following explicit rule:

```
< 3/5 mins, 5/15 mins, 15/hour, 25/day, 100/7 days >
```

Any number of threshold entries are allowed if a finer grain of monitoring is desired.

Two special threshold specifications are used to turn off monitoring or to generate alerts for all accesses. The specification '< >' denotes that no threshold is specified and turns off **Dynamic Activity Monitoring** for that particular rule. The specification '< 0 >' (zero) specifies that every access is to cause an alert.

3.3.2 Time Dependent Authorization

Recognizing that there will be instances where you wish to limit the time(s) of day or day(s) of the week that particular sites can access your network, we have implemented a flexible way to express this in an authorization file rule.

Time limits are enabled by inserting appropriate optional time range specifications in a rule in the authorization file. A time range specification is enclosed in square brackets '[ ]' and has the following syntax:

time-range [@ *time-range*] [; *time-range* [@ *time-range*] ...]

A *time-range* is a character string which specifies a start time and an end time, separated by a hyphen ('-'). White space is permitted within a time range, and start and end times may be any of the following formats, but the format must be consistent within a time range.

time of day Specifies an exact time of day in the form *hh:mm:ss meridian timezone*. Minutes, seconds, meridian, and timezone are optional. If no meridian - AM or PM - is specified, a 24-hour clock is used. Spaces may be used, but are not required. Timezone and meridian may be in either upper- or lower-case, with or without periods. Examples are: '08:15AM' (8:15 AM localtime) and '23 gmt' (11:00 p.m. GMT).

date A date is an absolute month and day, with optional year. Acceptable formats are a month name, day name or ordinal, or month, day, or year in numeric format *mm/dd/yy*. Day is a valid day name, number (1-31) or ordinal ('first' through twelfth,' excepting 'second' owing to ambiguity). The optional year is a numeric (i.e. 1991 or 91). Examples are: '05/15/93', 'January fifth', and 'Dec 31'.

Most common abbreviations (non-case-sensitive) for days and months are accepted, with or without a trailing period.

relative time Nonspecific dates/times are interpreted relative to the current time. Acceptable relative times are year, month, day, hour, minute, and second. Specifications follow the rules above.

The date is formed as follows: first, any absolute date and/or time is processed and converted. Using that time as the base, day-of-week specifications are added. Next, relative specifications are used. If a date or day is specified, and no absolute or relative time is given, midnight is used. Finally, a correction is applied so that the correct hour of the day is produced after allowing for daylight savings time adjustments.

To clarify the use of time ranges, we provide some examples below. An example of a time range that allows access any time Monday thru Friday follows:

```
[ Monday - Friday ]
```

Because the above time range does not indicate a specific date, it allows access on **any** Monday, Tuesday, Wednesday, Thursday, or Friday.

More complex date specifications include a qualifier–the '`@`' character–to specify both a day range and a time range. For example, to allow access via the **Eagle** during normal business working hours Monday through Friday:

```
[ Monday - Friday @ 8AM - 4:30PM ]
```

Multiple time ranges can be specified with the semi-colon. To allow access via the **Eagle** during normal business working hours and/or during the evening hours of Monday (only), use:

```
[ Monday - Friday @ 8AM - 4:30PM; Monday @ 4:30PM - 9:00PM ]
```

This facility is particularly useful when you have to provide short-term access to outsiders, such a service personnel. You do not have to worry about immediately deactivating their accounts when they are finished. This authorization facility prevents them from accessing your system before you are able to update your accounts list. In the next example, the users are limited to accessing your network during a specific 10-day period, at the end of which active connections are stopped, and further access is prevented.

```
[ December 1, 1993 12:00 - December 10, 1993 16:30 ]
```

3.3.3 Service Limits

As installed, the Eagle software blocks the passage of all network services into and out of your network, with the exception of SMTP (electronic mail). You can enable other network services to pass under the control of a powerful rules-based authorization file. customers want to be able to use the common ftp and telnet services, the ability to permit the passage of these services and to control who can use them, both inside of and outside of your Eagle-secured network. is a standard part of the software. All services, except `SMTP`, are denied to hosts not listed in your authorization file. You may want to provide additional services to some hosts and limit services to others. For example, you might only allow outside hosts to upload files via `ftp` to a single host on your internal network while allowing all inside nodes to download files from outside hosts. Similarly, you could deny a service altogether, such as `telnet`, except to designated hosts.

Service limit rules are enclosed in parentheses '( )' and may specify one or more of the limits in the table below.

Limit	Description
notelnet	The `telnet` service is disallowed to that machine.
noftp	The `ftp` service is disallowed to that machine.
ftp.getonly	Download (get)-only `ftp` service is allowed.
ftp.putonly	Upload (put)-only `ftp` service is allowed.
telnet.portnumber/service	Specify an alternative port number or service name for telnet connection auth
authenticate `telnet` and `ftp` users.	
noauth	allows access to that machine without requiring user authentication.
user	restrict `telnet` and `ftp` access into or out of the gateway to the users listed.
group	restrict `telnet` and `ftp` access into or out of the gateway to users belonging to the groups listed.
service	activate other network services defined in `passer.cf` (e.g. network news transport protocol (NNTP)). See Chapter **??** for more information on the generic service passer.

For example, to restrict users from `faraway.outbound.com` to file transfers only use this service limit:

```
faraway.outbound.com allow mymachine (notelnet)
```

Users may specify an alternative port number or symbolic service name (from /etc/services) on the destination machine. Authorization rules may contain restrictions on this facility. Examples follow:

```
mymachine allow faraway.outbound.com (telnet.all
```

This example allows users on `mymachine` to use telnet to specify all alternative port numbers on `faraway.outbound.com`. Other options include `telnet.none` (deny use of alternative ports), `telnet.finger` (limit use of alternative ports to the *finger* service, usuallly port number 79), `telnet.nofinger` (all use of all alternative ports *except* the `finger` service. In addition, you can specify any port number/service in an authorization rule, such as `telnet.3001`

By default, all users in the **Eagle**-protected network are authorized access all ports on remote hosts, while all remote hosts are limited to the standard `telnet` service.

Specify an alternative port in the telnet dialogue, as follows:

```
telnet firewall.xxx.com
Hostname: faraway.outbound.com portnumber/service name
```

Such `telnet` connections will use the `telnet` protocol for communication with the remote host if the service on the specified portnumber is a `telnet`-based service. Otherwise, the will merely pass characters between the two hosts. To force the connection to use the `telnet` protocol, use the following syntax (note the leading "-"):

```
telnet firewall.xxx.com
Hostname: faraway.outbound.com -portnumber/service name
```

By default, `telnet` and `ftp` users may have to identify themselves and enter a password to get into or out of a protected network. Use the `auth` service limit to specify a particular user authentication method. Use `noauth` to allow access to a machine without requiring user authentication. The syntax for specifying **auth** and **noauth** service limits is:

`auth = [` *method* `]`
`noauth`

where *method* is `gwpasswd` for *Gateway User Authentication*, or `ace` for Secure ID^{tm}, or both separated by a comma. If you list more than one method, each is tried until one succeeds. Refer to Chapter **??** for additional information on user authentication methods.

The **user** and **group** service limits refer to user authentication. These limits specify one or more users or groups that must identify themselves with a symbolic name and enter a password to get into or out of the protected network. Refer to Chapter **??** for details. The syntax for specifying **user** and **group** service limits is:

```
user = [ userlist ]
group = [ grouplist ]
```

where *userlist* is a list of one or more users (separated by commas) in the *Gateway User Database*, `/usr/adm/sg/gwpasswd`. *Grouplist* is a list of one or more groups (separated by commas) in the *Gateway Groups Database*, `/usr/adm/sg/gwgroup`. To negate the service limit, place an exclamation point at the start of the user list specification. If you wanted to permit `telnet` access only to users `sandi` and `steve`, for example, you would enter:

```
user = [ sandi,steve ]
```

To allow all users *except* `sandi` and `steve` to use `telnet`, enter:

```
user = ![ sandi,steve ]
```

The user list has precedence over the group list. For example if `steve` is in group `systems`, the following still allows `steve` to have access:

```
my.univ.edu allow myhost (user=[steve], group=![systems])
```

3.3.4 EtherGuard

Your authorization file may use rules to activate **EtherGuard** for additional protection against potentially vulnerable computers, including those with unsecure dial-up access. For an overview of **EtherGuard's** basic features, see Section **??**.

EtherGuard consists of two main components: the first component is more general and resolves host addresses for all machines. The second component secures specifically those personal computer hosts which have potential modem connections.

Since every network device on your network has an identifiable hardware address, your authorization file can map physical-addresses to IP-addresses for authorized hosts on your local network. The mapping is specified by entering a *source* host in the following format:

```
hostname=ab:cd:ef:gh:ij:kl allow host1, host2
```

The `'='` is followed by octet pairs which identify the host's network interface board's physical address. Use of the octet pairs signals **EtherGuard** to verify the host's mapping to a specified Ethernet address using the Address Resolution Protocol (ARP). This option only works for machines which are physically connected to the same network or subnet as the **Eagle**, since the ARP protocol cannot pass through network routers. **EtherGuard** cannot be used to identify and limit hosts on remote networks or subnets; the **Eaglet** network partitioner, however, can implement **EtherGuard** in other subnets.

Since PC's are inherently vulnerable (users can easily change the IP address, for example) we recommend that you include **Etherguard** rules for all PC's in your network.

EtherGuard also can protect your network against access from personal computers on the network which have dial-in modem connections. The format of the *source* hostname may be modified by adding a `'@'` character to the end of the PC's host name, and following it with the octet pairs, as illustrated below. The `'@'` character signals **EtherGuard** that the machine may have dial-in modem connections:

```
pchostname@=ab:cd:ef:gh:ij:kl allow host1, host2
```

EtherGuard uses special authentication for such a host, provided the PC is running *Raptor Systems'* modified, secure version of the NCSA `telnet` software. This software checks for a live modem connection and denies access if one is found. Thus, an outsider dialing into the PC would not be able to use the PC's network software to access the rest of your network or outside networks. PC's which do not have the modified NCSA `telnet` software are not granted network access when they are specified using `'@'`.

3.3.5 Eaglet

Your **Eagle** authorization file may contain rules for any **Eaglet** network partitioners on your network; the **Eaglet** itself does not store an authorization file on disk. Rather, the **Eaglet** receives its authorization information in encrypted form from the **Eagle** and maintains it in memory tables. The **Eaglet** 'knows' where the **Eagle** is on the network through a special file, `eagle.eyrie`, which contains a single entry which is either the hostname or the *IP* address for the **Eagle**.

Entries for **Eaglets** in the `gateway.cf` authorization file begin with the **Eaglet's** hostname and use the following format:

```
eaglethostname:  <source...> allow <destination...>
```

The **Eaglet** follows only those rules in the authorization file which apply to it. Other rules are ignored.

3.4 Macros and Comments

The authorization file is preprocessed with `cpp` (see your UNIX system manual pages for `cpp(1)`), a simple macro processor. This preprocessing allows you to take advantage of a useful construct, definition and replacement.

As in program source code, for example, the `#define` preprocessor command causes a *name* (identifier) to become defined as a macro to the preprocessor. What follows is the *body* of the macro. Where *name* is used in the authorization file, *name* is effectively replaced by a copy of *body*. If the macro is defined to accept arguments, then the actual arguments following *name* are substituted for formal parameters in *body*.

An example `#define` preprocessor command follows:

```
#define MYNET 192.24.345
```

This definition can be used in authorization rules in place of the more cumbersome *IP* network designation. If you defined the above macro in your authorization file, you could create rules such as the following:

```
.industry.com allow MYNET
```

Your authorization file may also contain comments. Comments begin with the characters '`/*`' and end with the first subsequent occurrence of the characters '`*/`.' Comments may contain any number of characters and extend over more than one line. Cooments are always ignored. A comment looks like this:

```
/* This is a comment */
```

Finally, authorization definitions or rules can be continued on the following line by appending a backslash \ to the current line:

```
alpha, .industry.com allow 192.48.145.1, \
192.24
```

3.5 Hierarchy of Authorization Rules

Rules in the `gateway.cf` configuration file have the form:

source **allow/deny** *destination*

A rule line is selected by the closest match to the incoming host first, and then by the closest match to the outgoing host. Thus, a rule line with an exact match on the incoming host and a general match on the outgoing host supersedes a rule line with a general match on the incoming host and an exact match on the outgoing host. If more than one line matches equally well, then the first rule encountered is used.

A blank incoming or outgoing host matches all hosts. This is the most general match possible. This can also be expressed as `0.0.0.0`, or just `0`. Normally, the first line of the configuration file is:

deny

This rule denies access to everyone. Subsequent rules define exceptions which permit or deny access in specific ways. Here is an example to illustrate this:

deny
`.xxx.com` **allow** `.yourdomain.com`
`.xxx.com` **deny** `securemachine.yourdomain.com`
`goodguy.xxx.com` **allow** `securemachine.yourdomain.com`

These rules could have been listed in any order. The first rule says to *deny all access unless overridden by another rule.* The second line permits all machines whose official host name ends with `.xxx.com` (i.e., all machines in the `xxx.com` domain), to access all machines in the `yourdomain.com` domain. The third line disallows access to the specific machine `securemachine.yourdomain.com` from the `xxx.com` domain. Rules two and three allow machines in the `xxx.com` domain acces to all `yourdomain.com` machines except `securemachine`. The last rule specifically allows `goodguy.xxx.com` access to `securemachine`.

3.6 Example Authorization Rules

For clarity, here are several examples depicting typical rule entries in a `gateway.cf` file.

In our first example, a host named `export`, all machines in the `.xxx.com` domain, and all machines on the network numbered `192.92.2` have full access to the machine named `mine` or the domain named `.yyy.xxx.com`, but are denied `ftp` access to `yours`. In addition, the `192.92.2` network applies a subnet mask in which five bits are used for hosts, and the remaining three bits denote subnet addresses. The configuration entry looks like this:

```
export, .xxx.com, 192.92.2&5 allow mine, yours(noftp), .yyy.xxx.com
```

In the next example we illustrate time limits. All machines from educational sites (`.edu`) are permitted `ftp` read-only access to the hosts in the `.xxx.com` domain from 9am to 5pm Monday through Friday, and noon to 4pm on Saturdays; telnet access is denied. The configuration entry would look like this:

```
.edu allow .xxx.com[Mon-Fri@9am-5pm;Sat@12pm-4pm](notelnet,ftp.getonly)
```

The following example shows how to change dynamic thresholds from the default. It states that that host `marcus.com` may try to connect to network `localnet` (which is defined in `/etc/networks`) up to 15 times per hour, and up to 50 times per day. This overrides the normal maximum connect limits for the period. More than 15 attempts during an hour, or more than 50 attempts in a day, will generate an alarm.

```
<15/hour, 50/day> marcus.com allow localnet
```

Next we illustrate how to set up the configuration file for an **Eaglet Subnet Partitioner**. (Note that the **Eagle** authorization file may contain rule(s) applying to one or more **Eaglets** in your network.) Suppose you want to limit users within a subnet so that they cannot access any systems outside their subnet, but you still want to allow people outside that subnet to log into those machines. The two lines below will only be used by the `Eaglet` named `badguyseaglet` to control network access by machines in the `.badguys.net` subnet. The first line states that *all* hosts (denoted by a blank *source* field) may access *all* machines (denoted by a blank in the *destination* field) in the subnet controlled by the **Eaglet**. The second line denies outgoing access for machines in the isolated subnet `.badguys.net`.

```
badguyseaglet:  allow
badguyseaglet:  .badguys.net deny
```

4 User Authentication

So far, this manual has described the **Eagle** as primarily a *network/host* authenticator. That is, your authorization rules allow or deny access based on hostname or network name, but do not authenticate *individual users.* Users on authorized hosts can go in and out of the protected network without identifying themselves. You can, however, have the **Eagle** identify and authenticate *who* is using the system. These users must identify themselves with a user name and a password or by SecurID unique key and passcode. This authentication is *in addition to* any other form of authorization/authentication required by the node being accessed or by the application being executed.

Users requiring password authentication must identify themselves by a user name and a password at the Gateway node. The name identifies the user and the password authenticates the user. If the gateway name or password is not valid, access thru the gateway is denied and the connection attempt logged. If the user is authenticated and all other conditions of the authorization rules are met, the connection is completed and the user information is recorded in the log. In addition to recording where a connection request is initiated, *who* requested the connection is now known.

4.1 User Authentication Methods

If you choose to authenticate users, you have two choices: (1) the *Gateway User Authentication* described in the next section or (2) Secure ID^{tm} from Security Dynamics Incorporated. The former method is described in the following sections. See your separate Secure ID^{tm} documentation for information about using the latter method.

4.2 E

agle User Authentication

The **Eagle**'s user authentication is based on separate *user* and *group* authentication, similar to UNIX user and group authentication.

4.2.1 Gateway User

Eagle] user authentication is based on the presence/absence of the *Gateway User Database* file `/usr/adm/sg/gwpasswd`. If this file does not exist, no user authentication is attempted. The *Gateway User Database* file contains usernames and encrypted passwords. In the presence of the *Gateway User Database*, anyone who wants to get into or out of the protected network must first identify himself and provide a password, with the exception of authorization rules containing the `noauth` flag. `Gwcontrol` checks this information by searching the *Gateway User Database*. If the name and password are verified the connection is allowed; otherwise access is denied. Either way the user name is logged.

With user authentication enabled, the Gateway responds to attempts to use `telnet` or `ftp` by first checking the authorization file for a rule matching the hosts involved. If a rule exists and the `noauth` flag is present, the connection proceeds normally. (Of course, if there is no matching rule, no connection is permitted.) Lacking the `noauth` flag (in the applicable rule), a valid Gateway username and password is required before the connection is permitted. Also, access can be limited to a specified set of groups and users with the `user` or `group` service limit in the authorization rule. See Section ?? for more details.

The *Gateway User Database* information is stored in `/usr/adm/sg/gwpasswd`. This file has a very flexible structure for each line. Each line, or *account entry*, has five fields separated by colons (:). If an account entry is missing a field, that account is ignored. Each entry in the `/usr/adm/sg/gwpasswd` file has these fields:

Field	Description
gateway user name	An alphanumeric user name.
encrypted password	
user id	The user id uniquely identifies every user with this number.
group id	Unlike the user id, the group id can be shared with other users. The id should correspond to an entry in the `/usr/adm/sg/gwgroup` file which provides a group name.
comment	Although this field may be empty, it most commonly contains identifying information about the account.

An example gateway user entry follows:

```
sandi:BhNqayKxwq2r:20:100:Sandi Tennyson
```

Every gateway user must have an individual account: if users are permitted to share accounts, the *accountability*, the ability to determine who did what, is lost. The account entry provides the user identification information when the user wants into or out of the protected network.

Each gateway user entry requires an encrypted password. Eagle encrypts passwords using is a specially enhanced version of the Data Encryption Standard (DES), vastly more secure than the standard UNIX system password encryption. Passwords are set and/or changed with the `/usr/adm/sg/gwpassword` program, which must be run by the super-user on the G Box console. (Individual users cannot change their Gateway password.) Follow `gwpassword` with a username. `gwpassword` prompts for the new password. The password is not displayed to prevent others from viewing it. It is requested twice to verify it was entered correctly. The example below shows the normally invisible password exchange.

```
% gwpassword sandi
Changing password for sandi.
New password:  doggie16
Retype new password:  doggie16
```

If the two passwords you type don't match, the password remains unchanged. New passwords should be at least five characters long, if they combine upper-case and lower-case letters, or at least six characters long if in monocase.

4.2.2 Gateway Group

A gateway group is a set of one or more users who are associated for some reason. The primary use of a gateway group is to simplify the writing of authorization rules which can apply to groups of users. The *Gateway Group Database* information is stored in `/usr/adm/sg/gwgroup`. Each group entry in the `/usr/adm/sg/gwgroup` file has these fields:

Field	Description
gateway group name	An alphanumeric group name.
comment	Although this field may be empty, it most commonly contains identifying information about the group.
group id	The numerical group id can be shared with other users.
user list	A list of users separated by commas.

An example gateway user entry follows:

```
trusted:Security Staff:100:sandi,bill,steve
```

Only user `root` on the G Box can make changes to the *Gateway User Database* or *Gateway Group Database*.

4.2.3 P

recedence of Rules in User Authentication

To determine whether to authenticate a user or which method to use, `Eagle` exercises these rules in the order shown until one is true:

1. Use any method specified as a rule service limit with `auth` or do not authenticate users if `noauth` is specified. Refer to section ?? for a description of these limits.
2. If the connection is incoming and file `/usr/ace/sdconf.c` or `/var/ace/sdconf.rec` exists, use Secure ID^{tm}.
3. If the connection is incoming and file `/usr/adm/sg/gwpasswd` exists, use *Gateway User Authentication*.
4. If the connection is outgoing and file `/usr/adm/sg/gwpasswd` exists, use *Gateway User Authentication*.
5. If the connection is outgoing and file `/usr/ace/sdconf.c` or `/var/ace/sdconf.rec` exists, use Secure ID^{tm}.
6. Authorized `telnet` and `ftp` users can go in and out of the protected network without identifying themselves.

4.3 TELNETD and FTPD

With user authentication enabled on your Eagle system, the `telnet` and `ftp` dialogue users engage in are expanded to include the authentication process, with users being prompted for the gateway username and the gateway password, in addition to the information normally required.

In the examples below we assume that user authentication is enabled and the following line is in the Eagle's configuration file (see Chapter ??):

```
faraway.outbound.com allow mymachine
```

This permits users on `faraway.outbound.com` (outside the Eagle-protected network) to connect to `mymachine` (inside the network), but requires an additional gateway username and password. In these examples, the name of the Eagle is `firewall.xxx.com`.

4.3.1 T

elnet Dialogue With User Authentication

Figure ?? illustrates the dialogue for connecting from `faraway` to `mymachine`.

Note the differences between this example and the standard `telnet` connection dialogue, as well as the differences between this dialogue and the standard Eagle `telnet` dialogue..

1. The user specifies the Eagle Gateway as the destination host, not the true destination host.

2. The Eagle Gateway prompts for the true destination hostname

3. The Eagle Gateway prompts for the Gateway username and password before connecting to the destination host

4. The destination host prompts for a username and password before permitting the login.

The user is now logged in as user `username` on `mymachine` via *passthrough* `telnet`. The Eagle will continue to transparently pass characters between the two systems until the user logs out.

faraway% `telnet firewall.xxx.com`	*Connect via the* **Eagle** *for passthrough*
Trying 192.92.2.1...	*IP address of the* **Eagle**.
Connected to firewall.xxx.com.	
Escape character is ']'.	
This is a private network. Enter hostname.	*Message in the motd file.*
Hostname: **mymachine**	
Enter Gateway Username: gw-user	*Gateway username.*
Enter Gateway Password:	*Password not echoed to screen.*
	Verifying connection.
SunOS UNIX (mymachine)	*Host banner message.*
login: **username**	*Name to login as.*
Password:	*Password not echoed to screen.*

Figure 4.1: Sample User Authenticated TELNET Dialog

.

4.3.2 f

tp Dialogue with User Authentication

The logon dialogue for user-authenticated `ftp` differs similarly. (The following example uses the same authorization rule shown above.)

Note the differences between this example and the standard `ftp` connection dialogue, as well as the differences between this dialogue and the standard Eagle `ftp` dialogue.

1. **The user specifies the Eagle Gateway as the destination host, not the true destination host**
2. **The Eagle Gateway prompts for the true destination hostname and username, in the format username@hostname**
3. **The user also appends the Gateway username on the same line as the destination hostname and username (omission of this optional means the same username is to be used for both systems)**
4. **The Eagle Gateway prompts for the user's password on the destination host.**

faraway% `ftp firewall.xxx.com`	*Connect via the* **Eagle** *for passthrough*
Connected to firewall.xxx.com.	
220 firewall FTP server ready.	
Name: **username@mymachine gw-user**	*User and machine to access. followed by the gateway username.*
331 Password required for username.	
Password:	*Password not echoed to screen.*
332 Gateway password required to authenticate this request.	
Password:	*Password not echoed to screen.*
230 User username logged in.	
ftp: **get afile**	*Allowed to get (download) files.*
150 ASCII data connection for afile (9999 bytes).	
226 ASCII Transfer complete.	
9999 bytes received in 0.10 seconds (99 Kbytes/sec)	
ftp:	

Figure 4.2: Sample User Authenticated FTP Dialog

.

5. The Eagle Gateway prompts for the Gateway user's password

The Eagle will continue to screen commands until the user logs out.

When the connection to the destination host is closed, the service daemons will re-connect to `gwcontrol` to report that the connection is no longer there. `Gwcontrol` then removes that connection from the active list and updates the log.

5 Generic Service Passer

By default, the Eagle allows only the `ftp` and `telnet` network services to pass through a protected network, subject to rules you place in your authorization file. There are many other network services, some of which you might want to allow through your G Box. The Generic Service Passer allows you to do so by importing (or exporting) a service from a computer on one side of the Eagle to a machine on the other side. A network service is set up by someone elsewhere to provide access to their machine for a specific purpose. Some services are informational (WAIS, WWW, gopher, archie, etc.), some are recreational (MUDs, IRC, etc.) and of course, there is USENET news. There are more services and servers than we can describe here. A summary of some services may be obtained via anonymous `ftp` from the host `ftp.rpi.edu`, by copying the file `pub/communications/internet-tools`.

As a security measure, the Eagle disallows direct access between a local computer and an external computer. The Generic Service Passer is a mechanism you can use to permit specific services into or out of your network, through what might be called a 'proxy' connection through the Eagle to the real service machine.

There are two steps needed to set up passing a service. First, add the service to the Passer's configuration file `/usr/adm/sg/passer.cf`, on the G Box. This specifies where the real server machine is, and what service name to use to get to it. The second step is to modify the gateway configuration file `/usr/adm/sg/gateway.cf` (on the A Box) to specify which machines may have access to the service.

To add a service to the *passer* configuration file, enter a line of the form:

> *service serverhost*
> *service serverhost/serverport* +
> *service serverhost* **countdeny**

Serverhost names the server computer by its hostname or IP address. *Service* is either a service name from the /etc/services file or a *portnumber/protocol* pair for the service (where *protocol* is either *UDP* or *TCP*). *Serverport* allows you to specify the port on *serverhost* to connect to, otherwise it is the same port as *service*. *Serverport* is used in situations where the service is provided on a port number other than the standard one, or when you want to receive the service on a different port. The + option (TCP only) tells the Generic Service Passer to write as the first packet of a connection the message:

```
IPaddr port \n
```

This is meant to let the real server know what the real client's Internet address and port are (the server will probably need to be set up specially to understand this). The *n* stands for an ASCII linefeed character (hex 0A).

The *countdeny* argument increments the sliding time window (see Dynamic Activity Monitoring) only on *access denied* events.

To allow the use of a *service*, edit the gateway configuration file on the Authorization machine. Add the *service* to the option list for a rule which allows or denies the connection. For example, if the line in the Passer config file was:

```
service servicehost
```

and you wanted only mymachine to use that service, the line in the gateway config file would be:

```
mymachine allow servicehost(service)
```

In practice, mymachine connects to the Eagle instead of servicehost, and specifies the desired service. The Eagle verifies that the connection is allowed. If the connection is allowed, the Eagle then connects to servicehost. From that point the Eagle simply relays characters between mymachine and servicehost. For *UDP* services in which there is no 'connection' per se, a pseudo-connection is used which will shut down after a minute of inactivity. All connections are logged.

More examples follow, but first a caution: network service clients may have security holes which could allow someone to break into your network through the service passer. While the chance of such a breach is remote, the potential remains because you can allow any service to pass through the service passer. You can minimize the risk of break-in by running up-to-date system software and limiting these connections to trusted machines. Otherwise, you can defeat the purpose of the Eagle, and Raptor Systems cannot guarantee the security of your system.

The following example sets up the local machine `newshost` to receive USENET news on its *NNTP* port. When a remote machine wants to send news, it connects to the Eagle at the *NNTP* port (119/tcp). The Eagle then connects to the same port on `newshost`, and copies the data between the two connections. The entry in the configuration file (passer.cf) would look like this:

```
nntp newshost
```

An authorization rule for this entry which allows *remotenews* to access the service would be:

```
remotenews allow newshost(nntp)
```

Please note that this configuration allows incoming USENET news only. Articles posted by local users will appear on your internal news server, but will not be propagated to the outside world. See /bf Raptor Systems Technical Note #2 for an alternative mechanism which enables bidirectional NNTP service.

In the next example, we set up *UDP* port 1123 on the Eagle to allow local client machines to use the *Network Time Protocol* service (123/udp) on the server machine `mimsy.umd.edu`:

```
1123/udp mimsy.umd.edu/123
```

Note that in this example, we changed port numbers from the default port 123 to 1123 on the Eagle.

The warning about enabling potentially insecure services and/or access by untrusted machines bears repeating. Your purpose in purchasing the Eagle was to protect your network. Unwise use of the Service Passer could defeat the purpose, and Raptor Systems cannot be responsible for security breaches caused by such use.

6 Message Log

6.1 Overview

A log is maintained of all messages, describing the opening, closing and denial of all connections. Eagle system errors are also sent to the log, as is diagnostic information if suspicious activity is detected. This represents an audit trail of the activity on your gateway. It provides valuable clues in the event there is an attempted security breach of your network.

Various activities produce messages which are saved in a log file on your G Box. The individual messages are described in Appendix ??. The messages are posted through `syslogd` to `/usr/adm/sg/logfile`. Refer to `syslogd(8)` for a description of the system log facility. At the end of each day the current logfile is archived. These logs can be perused anytime or loaded into a database for processing. Sample log entries are shown in Table ??. The next sections describe the programs that process the message logs.

6.2 Change Log

At the end of each day, the current log file is moved to the `/usr/adm/sg/oldlogs` directory. The file is named `logfile.`*YYMMDD*. Where *YYMMDD* represents the year, month, and day the log was created.

There are two main components to the change log facility:

- The `changelogfile` shell script (found in `/usr/adm/sg`)
- The `changelog` executable program

Sep 30 00:00:02 wolf changelog: 108 starting new log file
Sep 30 03:14:39 wolf : 101 Eagle Network Isolater starting up
Sep 30 03:52:26 wolf : 105 telnet[211]: connection for localhost to bear
Sep 30 03:52:37 wolf : 103 telnet[211]: closing connection
Sep 30 03:52:48 wolf : 105 telnet[222]: connection for wolf to bear
Sep 30 03:53:10 wolf : 103 telnet[222]: closing connection
Sep 30 03:53:42 wolf : 102 Shutdown command received, gwcontrol quitting
Sep 30 03:53:48 wolf : 101 Eagle Network Isolater starting up
Sep 30 03:54:24 wolf : 201 telnet[247]: access denied for wolf to wolf
Sep 30 03:54:33 wolf : 105 telnet[253]: connection for localhost to bear
Sep 30 03:54:41 wolf telnet[253]: 110 user steve authenticated on connection from localhost to bear
Sep 30 03:54:45 wolf : 103 telnet[253]: closing connection
Sep 30 03:55:10 wolf : 105 telnet[274]: connection for localhost to bear
Sep 30 03:56:00 wolf telnet[274]: 202 access denied for sysdiag from localhost to bear
Sep 30 03:56:10 wolf : 103 telnet[274]: closing connection
Sep 30 04:02:39 wolf gwpassword: 203 User steve password changed
Sep 30 04:26:09 wolf : 105 ftp[391]: connection for localhost to bear
Sep 30 04:26:12 wolf ftp[391]: 110 user steve authenticated on connection from localhost to bear
Sep 30 04:26:32 wolf : 103 ftp[391]: closing connection
Sep 30 05:31:35 wolf : 104 re-read of new config file successful
Sep 30 06:13:26 wolf : 105 telnet[29654]: connection for wolf to bear
Sep 30 06:13:32 wolf : 103 telnet[29654]: closing connection
Sep 30 06:14:05 wolf : 105 telnet[29676]: connection for wolf to bear
Sep 30 06:14:05 wolf : 501 access from wolf to bear [rule line 6]: over 2 tries in 1 minutes
Sep 30 06:14:06 wolf traceroute: 106 to wolf: wolf(192.5.57.2)
Sep 30 06:14:20 wolf : 103 telnet[29676]: closing connection
Oct 1 00:00:01 wolf changelog: 107 closing log file

Table 6.1: Sample log entries.

`changelogfile` is executed daily at midnight by `cron`. Refer to `cron(8)` for more details. It runs `changelog` to move the daily log to the `oldlogs` directory. You can edit `changelogfile` to perform special handling of the daily log. This includes mailing the log file to an account on another machine, printing it, flattening it to load into a database, or deleting older log files to conserve disk space. Refer to the comments in `changelogfile` for details on how to affect these features.

6.3 Flatten - Reading the Message Log

The program `flatten` formats the message log file to allow it to be more easily read and processed by a database management system or by operating system utilities (sed, awk, etc.). `Flatten` resides in the directory `/usr/adm/sg`. It reads the free format records found in the Eagle log, reformats the records in a fixed field format, and prints them on standard output. When completed, it prints on standard error the number of records skipped due to badly formed log messages.

In the output, each field is of variable width and is separated from the next by one space. The last field is terminated by a `newline` (ASCII linefeed). The description of the fields below are given in the same order as they appear in the output.

date The date of the log message is given in the form *MM/DD/YYYY*, so that August 1, 1993 will show up as 08/01/1993.

time The time of the log message is given in 24-hour clock style in the form HH:MM:SS. Example, 08:15:01.

service or program This tells what generated the message. If the ID (described next) is zero, this is a program like `gwcontrol` or `passer`. If ID is non-zero, it is a service like `ftp` or `telnet`.

ID This is the identification number of a connection if non-zero. If zero, the message does not describe a connection.

message level This tells at what severity level this message is. Values are: `Info`, `Notice`, `Warning`, `Error`, `Alert`, `Critical`, `Emergency`, or `Unknown`. Refer to Appendix ?? for a description of these severity codes.

message id This gives the unique message id number that every message in the Eagle log file has. Example: *101* for `Eagle Network Security Management System starting up`. See Appendix ?? for a description of log messages.

argument 1

argument 2

argument 3

argument 4

argument 5 These five fields are variable parts of the log message broken out into separate values. The values are different for each separate message. An argument that is not used by the message will have a value of 0, in order to keep the same number of fields on an output line. Only the following message id's have any arguments:

- **105** a1=incoming host, a2=outgoing host
- **106** a1=traced host
- **201** a1=incoming host, a2=outgoing host
- **310** a1=host name, a2=host *IP* address
- **311** a1=host name
- **312** a1=host name
- **406** a1=audio file name
- **417** a1=host name, a2=port number
- **501** a1=incoming host, a2=outgoing host
- **502** a1=host name, a2=ethernet address
- **503** a1=host name, a2=host *IP* address
- **605** a1=program name

log message *(optional)* This field contains the entire text of the log message as specified by ID. Note this field has a variable length and may contain spaces. It is terminated with `Newline` character (ASCII Linefeed). Use `-n` as a command line argument to exclude this field from the flattened record.

7 Notify

This is a critical feature of the Eagle. *Please* take the time to develop a security plan including who you will notify if problems occur, and who their backups will be.

`Notifyd` works closely with the messaging capability of the Eagle. There are various levels of criticality of messages which can be assigned to any one or more types of message transport. If you wish to be notified of any alerts from the Eagle, you must include the proper entry in `/usr/adm/sg/notify.cf`, on the G Box. An entry consists of three or more fields separated by colons: message severity, time range, transport, and one or more optional arguments. The general format of the configuration file is:

severity : time range : transport [: arg 1 [: arg 2]]

The message severity indicates which types of messages you are interested in receiving. See Appendix ?? for a description of the different message types. Note that you can specify more than one message severity or a range of severities. To specify more than one message severity, separate each one with the | character. To specify a range, use `UpTo(`*severity*`)`. This includes all messages from *severity* up to and including the highest severity (Emergency).

You can also restrict when you are notified of suspicious activity to particular hours of a day, or days of a week, etc. The syntax for specifying time range in the configuration file is the same as that described in section ??.

The transports are:

- Audio - generate a sound on the Eagle
- Mail - send electronic mail

- Page - contact someone on pager
- Fax - send a fax
- Client - execute a client program

For the audio transport the optional fields are the name of a sound file followed by a volume (0-100). For the electronic mail transport, only the recipients are required (i.e. `user@hostname`). To correctly page or fax someone, you need a name and a phone number. The phone number is passed directly to the modem and must follow the Hayes standard (refer to your modem manual).

The client transport requires a program name. You may define your own program(s) for this function. When the program is executed its arguments are the message priority, the message, and any optional fields you supply.

To continue a particular field to the next line, end the current line with a `backslash` character.

An example notification configuration file follows:

```
Emergency : Sunday-Saturday : Fax : Pat Wales : 5551700
Critical|Alert : Monday-Friday : Page : Tom Ho : 9,5551692,,1234#;
UpTo(Warning) : Monday-Friday @ 8AM-4PM : Mail : hause@fantasia
UpTo(Warning) : Sunday-Saturday : Client : dispatch : 6668DE
Alert : Sunday-Saturday : Alarm : alarm.au : 50
```

Here we see that Pat Wales is notified via a fax of any emergency any time during the week. Tom Ho gets notified of critical or alert events via his pager with the digits '1234' displayed. Hause will receive a mail message from any event priority from Warning to Emergency, but only during normal working hours during the business week. Anytime a Warning to Emergency occurs, the client program `dispatch` is executed with the arguments: message priority; message; and `6668DE`. Finally, an audio alarm will sound on the gateway machine whenever an Alert message is issued.

For a notification to be sent to a pager, a V32.bis modem must be attached to your system.

For a notification to be sent to a fax machine, a Class 2 fax modem must be attached to your system.

There is no need to have two modems on the Eagle. The notification system assures that at no time will there be an attempt to use the fax and the pager simultaneously.

8 Eaglet

The Eaglet subnet partitioner is used in situations where further isolation is required within the local area network. The same type of computing hardware is recommended for the Eaglet as for the Eagle; however, no additional coupled authorization box is required. The Eaglet, as the name suggests, acts as an 'offspring' of the Eagle, receiving guidance rules (the authorization file) from the parent Eagle. This is done either upon startup of the Eaglet or following changes that have been made to the authorization file; the Eagle 'force feeds' changes to Eaglets. The latter is an automatic process: any time a configuration change is made, `gwcontrol` communicates this change to each Eaglet (specified in the file `/usr/adm/sg/eaglets`). The authorization rules are sent over the network in an encrypted form to prevent reading them with network sniffers or other similar mechanisms.

While each Eaglet holds in memory (not on disk) a copy of the authorization file, only those entries in the file which specifically refer to that Eaglet apply to the subnet being guarded. Thus, multiple Eaglets can run in your network, with each one's security enforced. Therefore, someone outside a protected subnet is unable to 'sniff' or snoop confidential information that is being restricted to that subnet. For example, there is no need for someone outside of the accounting department to ever be given the opportunity to indiscriminately listen to accounting data packets for which they do not have a legitimate need. Ethernet sniffing is extremely hard to manage on an enterprise wide basis because of the broadcast nature of the transmissions. Some vendors have accomplished this through smart (and expensive) bridges, but there is nothing that they can do if the communication is via X.25 or Token Ring network media.

As described earlier, the authorization file may contain rules applying to Eaglet(s) in the network. Here is the syntax:

```
eaglethostname:  source allow destination
```

Note that the Eaglet's hostname precedes the *source* host in this rule, with the colon as a separator.

The gwcontrol program on the Eaglet will report and log activity with respect to the subnets to which it is connected. The same full set of rule and notify features which apply to the Eagle may be rules for the Eaglet.

A Software Installation

Normally the Eagle Network Security Management System software comes pre-loaded on your computer. However, there may be times when you will need to load (or reload) it from the distribution tape. The software comes on 150MB (or 60MB) 1/4-inch tapes (QIC) in `tar` format. The software needs to be loaded on both the *Gateway* machine and the *Authorization* machine. In the instructions below, the files are loaded into the `/usr/raptor` directory. You can specify any other directory if you want; if you do so, be sure to substitute that directory name for `/usr/raptor` in each of the steps below.

A.1 Installation Worksheet

Before beginning the software installation, take a few moments to fill out the Installation Worksheet (see figure ??). This worksheet has spaces for important information about your Gateway and Authorization machines.

A.2 Installing the Eagle Software

Here are step-by-step installation instructions. In most cases, each step needs to be carried out on both the G box and the A box. In this example, assume that the G box is named *gateway* the A box is named *author* and a third machine with a tape drive (if needed, should your systems not have their own tape unit) is named *tapehost*. (Once operational, the A box is not connected to your local network, but you may need to have it connected for the installation if it does not have a local tape drive.)

Gateway Machine

- Hostname of first network interface (e.g., gateway-outside):
- IP Address of first network interface (e.g., 192.23.45.67):
- Hostname of second network interface (e.g., gateway-inside):
- IP Address of second network interface (e.g., 192.23.45.68):

Authorization Machine

- Hostname of Authorization Machine (e.g., author):
- IP Address of Authorization Machine (e.g., 192.23.45.69):

Figure A.1: Configuration Worksheet For Eagle Installation

1. Identify your tape drive device name. On Sun SparcStations, the tape drive is normally `/dev/rst0`. On IBM RS/6000 workstations, the following command will list your tape devices: `lsdev -C -c tape -H`. AIX tape device names are of the form `/dev/rmtX`. On HP-UX, tape device names are of the form `/dev/rmt/X`. You will substitute the device name of your tape drive in commands below for device_name; be sure to include the leading `/dev`.

2. Create the directory `/usr/raptor`. Make sure that the disk partition has at least 10 megabytes free.

```
mkdir /usr/raptor
cd /usr/raptor
```

3. Load the software. There are three ways to load it, depending on your hardware configuration and network setup.

   - From a directly attached SCSI tape drive, enter:

     ```
     tar xf device_name
     ```

   - Across the network from another machine with a tape drive using `rsh` (on HP-UX, substitute `remsh` in the examples below for `rsh`). The hostnames of your G and A boxes will need to be in that machine's `/.rhosts` file for this to work. In this example, *tapehost* would have the following `/.rhosts`:

     ```
     gateway root
     author root
     ```

     See your manual page for `hosts.equiv(5)` for more information, including important warnings, about the `/.rhosts` file. Enter the following command to load the software from the remote tape drive, substituting the correct device name:

     ```
     rsh -n tapehost dd if=device_name bs=20b | tar Bxf -
     ```

   - Across the network from another machine with a tape drive using another method.

     (a) Log onto *tapehost*.

     (b) Change to a directory whose file system has approximately 6 megabytes of free space.

     (c) Type:

     ```
     dd if=device_name of=Eagle.tar bs=20b
     ```

     (d) Type:

     ```
     ftp gateway
     ```

(e) Log in as root, giving the appropriate super-user password.

(f) Type:

```
binary
cd /usr/raptor
put Eagle.tar
quit
```

(g) Repeat steps ?? through ?? with *author* as the destination for the ftp file transfer.

(h) Still logged into *tapehost* type:

```
rm Eagle.tar
```

(i) Log out of *tapehost* and go back to *gateway*, logging in as root.

(j) Type:

```
cd /usr/raptor
tar xf Eagle.tar
rm Eagle.tar
```

(k) Login to *author* as root and repeat step ??.

4. Install the Eagle software.

   Eagle software is installed in steps. *Step 1* is performed on the *gateway* machine, and *Step 2* is performed on the *author* machine. If you are installing a single-machine Eagle system, run *Step 3*. As with the above steps, you must be logged in as root.

   To start the installation, type:

   ```
   cd /usr/raptor
   ./Install
   ```

   You then choose the steps you want, and answer the questions. These questions concern what hostnames you have chosen for the Gateway and Authorization machines and their IP addresses. Refer to your Configuration Worksheet for this information. Once you have answered these questions, the install procedure will complete automatically. This will take a few minutes, particularly if you are installing on a SparcStation, as the install builds a new operating system *kernel* for PPP support.

   As a final step, create a login password for the special user `display` was created by the installation script. (Use the `passwd` program to do this.) This user operates with super-user authority on the system, so be sure to keep this password as secure as you would that of the `root` user.

5. Configure the Eagle software.

Eagle is configured by editing a few text files with any ASCII text editor, such as Sun's `textedit`, IBM's `INed`, or the generic UNIX editor `vi`; you may also use any other text editor you may have installed, such as `emacs`. The format and descriptions of these files are described throughout this manual. These files are located in the `/usr/adm/sg` directory:

File Name	Description
gateway.cf	This is the most important configuration file. It defines all permissions of the Eagle. It can only be edited on the A box, then reloaded into the `gwcontrol` program by typing `C` into the `display` program.
notify.cf	This file says who to notify when certain messages happen. You can choose a transport from among mail, fax, pager or audio alarm. This file resides on the G Box.
passer.cf	This is the configuration file for the Generic Service Passer for passing through services to other machines. This file resides on the G Box.
changelogfile	This shell script is run automatically at midnight to close the day's log file and re-open a new one. By default, it moves the old log into the directory `/usr/adm/sg/oldlogs`. A different disposition for the log can be made by editing this script. This file resides on the G Box.
gateway_motd	This is the message that users will see when they telnet to the Eagle. This file resides on the G Box.
ftp_motd	This is the message that users will see when they ftp to the Eagle. This file resides on the G Box.

6. Attach a null-modem RS-232 cable. On SparcStations and HP 9000's, connect the cable from serial port "b" of the G box to serial port "b" of the A box. On RS/6000's, use serial port #1 on each. If you already have your console terminal connected to this port, contact Raptor Systems for help in configuring your system.

   This cable forms a private, secure link between the G and A boxes, enabling the Eagle to access the authorization rule database in a secure fasion.

7. Reboot the A box then the G box.

The software is pre-configured for the CPU boards of your computers, and cannot be installed on any other system. If you need to replace a CPU board or install the Eagle software on another system, contact *Raptor Systems* for updated software.

If you have to change the IP addresses assigned to the ethernet ports (because, for example, you want to install the Eagle on a entirely different network), just re-run install *Step 2* for the G box and install *Step 3* for the A box.

A.3 Uninstalling the Eagle Software

When the Eagle software is installed, it modifies several existing files and renames the originals as *filename*`_eagle`. When the software is uninstalled, these original files are restored to their original location. To uninstall the software, execute step 6 in the Install script:

```
# /usr/raptor/Install
```

Note, if the Eagle software is installed and you plan to reinstall it, *uninstall* it first to restore your system to its original state.

A.4 Changes Made by the Eagle Installation

As noted in the previous section, installation of the Eagle software makes a number of changes to your system's configuration. Here is a summary of these changes.

A.4.1 Changes to /etc/inetd.conf and /etc/services

The `/etc/inetd.conf` and `/etc/services` files control which network services your computer system provides. Since many of these services are insecure, the Eagle installation disables those which are insecure. Examples include the `finger`, `lpd`, Network Information Services (NIS), and `rwhod` daemons. In addition to completely disabling several standard network services normally controlled by the `inetd` daemon, the Eagle installation also replaces the `ftpd` and `telnetd` daemons with secure versions provided by Raptor Systems. HP-UX users should note that the HP-specific inetd security file `/usr/adm/inetd.sec` will have no effect on services disabled in `/etc/inetd.conf` by the Eagle installation script; if the service is not

available at all, it cannot, obviously, be controlled by this secondary configuration file.

The following subsection describes important changes made by the installation to the Network File System services on your A Box and G Box.

A.4.2 Changes to the Network File System Services

Eagle makes important changes to the Network File System (NFS) configuration of your A Box and G Box with respect to NFS. *It is important that you understand these changes and how they ensure the security of your system–and that you not override these changes.* **First, the Eagle installation turns off NFS** *server* **services for your G Box. This step ensures that no other system–not even your A Box can access NFS services from your G Box and prevents access to the important information stored in your Eagle logfiles.**

Also, the installation *enables* **NFS server services on your A Box, but strictly limits access to them for security purposes. The key to the security of the Eagle system is the security of your A Box and its authorization database. This file is physically stored on the A Box and NFS-exported for reading by the Eagle software on the G Box. The permissions for this NFS mount limit access to the authorization file to the G Box alone, and that in** *read-only* **mode.** *These limitations are absolutely critical to the security of your system and should not be changed in any way for any reason.* **It is very tempting, for example, to make this NFS mount read-write for the G Box to enable the system administrator to make changes in the authorization file from the G Box console. Don't do this. To ensure the security of your system, you should make changes to the authorization file only from the console of the A Box, and not through any other means.**

B Log Messages

This appendix describes all the messages that may occur in the Gateway log file. Each message has an associated message number to help identify it. The first digit of the message number also gives the severity of the error. The message severity codes are:

1. Informational
2. Notice
3. Warning
4. Error
5. Alert
6. Critical
7. Emergency

See chapter ?? for details.

B.1 Informational Messages [severity Info]

Log messages in the range 100-199 are informational. They are standard operational messages which indicate the gateway is operating properly. The message number, the message text, and a description of these messages follow:

101 Eagle Network Isolator starting up

The gateway control program `gwcontrol` has been restarted.

102 Shutdown command received, gwcontrol quitting

`gwcontrol` received the command to stop all current connections and terminate. No new connections will be possible until `gwcontrol` is restarted.

103 *service*(*id*): closing connection

This message shows the orderly end of a connection. This should have a matching connection message (105) based on the *id*.

104 re-read of new config file successful

`gwcontrol` received the command to reconfigure itself and successfully executed it.

105 *service*(*id*): connection for *incoming* to *outgoing* (rule line *N*)

This message describes a connection made from the *incoming* host to the *outgoing* host for the *service*, i.e., `telnet` or `ftp`. The identification number for the connection is *id*. The line number within the configuration file is *N*.

106 traceroute: to *trace*: *host1* - ... - *host*n

This message is generated by the `traceroute` program. It lists the probable path through the network that packets between the **Eagle** and the *trace* host take.

107 changelog: closing log file

108 changelog: starting new log file

The log file is changed daily. At midnight, the `changelog` program renames the old log file to oldlogs/logfile.YYMMDD and a new log file is created.

109 passer: re-reading config file

`passer` checks every few seconds to see if its configuration file has been modified. If it has, it will apply the changes to its running configuration and display this message.

110 *service*(*id*): user *user* authenticated on connection from *incoming* to *outgoing*

The connection from *incoming* to *outgoing* required the user to authenticate himself, which was successfully completed. This message is only generated if the Gateway User Database has been set up.

111 *service*(*id*): *direction* file '*filename*' (*N* bytes) *Error*

FTP logs all file transfers with this message. The *direction* is either `retrieved` or `put` and *filename* is the name of the client side file. The number of bytes copied is also reported and any error is noted.

B.2 Notice Messages [severity Notice]

Log messages in the range 200-299 are notices. They occur when someone attempts to reach a machine and is denied. The message number, the message text, and a description of these messages follow:

201 *service*(*id*): access denied for *incoming* to *outgoing*

If the configuration file does not permit the *incoming* host to use the *service* on the *outgoing* host, this message is sent.

202 *service*(*id*): access denied for *user* from *incoming* to *outgoing*

The connection from *incoming* to *outgoing* required the user to authenticate himself, and failed. This message is only generated if the Gateway User Database has been set up.

203 gwpassword: User *user* password changed

The `gwpassword` program was run and the *user*'s password was changed in the Gateway User Database.

B.3 Warning Messages [severity Warning]

Log messages in the range 300-399 are warnings. Warnings occur when there is a recoverable error. The message number, the message text, and a description of these messages follow:

301 Internal warning: *message*

This message indicates an internal error to the **Eagle** software. The system can usually continue normally, but this could possibly signal that there is something else seriously wrong.

303 Warning: gwcontrol is already running

`Gwcontrol` checks to see if a copy of it is already running before continuing. If there is, it prints this message and exits.

304 passer: Line *N*: Warning: protocol mismatch, assuming *proto*

While reading its configuration file, `passer` found a mismatch in protocols (UDP vs. TCP) between the requested port and the port on the real server. This messages says that `passer` is assuming protocol *proto* for the service, and continues.

305 passer: Line *N*: Warning: ignoring multiple entry for *service*

While reading its configuration file, `passer` found more than one line that requires the same *service* port. `passer` ignores the second one and continues.

306 Warning: overlapping time range *time range*

In one of the rules, Time Dependent Authorization was used and the *time range* overlapped with a previous one.

307 Warning: config file not from Authorization machine

The gateway configuration file should be mounted from the Authorization machine. This message is shown when the config file that `gwcontrol` has read is not a mounted file. This could indicate someone was bypassing security by replacing the config file with a counterfeit.

308 Warning: can't lookup host *name*

The host *name* is mentioned in one of the configuration files, but it cannot be looked up to find its corresponding *IP* address. The system tries to continue without it.

309 Warning in config file (line *N*: *error*)

While reading the config file, `gwcontrol` came across a problem in the configuration file, but was able to continue reading. The one line that caused the problem is ignored.

310 *service*(*id*): *name ipaddr*: can't verify reverse address

In trying to verify the identity of host *name*, the *service* couldn't lookup the *ipaddr* returned by a previous successful lookup. This check is done to spot someone trying to spoof the *IP* address of another machine.

311 Warning: can't verify ethernet address for *host*

`Gwcontrol` tried to determine the hardware ethernet address of *host*, but received no answer.

312 *service*(*id*): bogus response to name lookup for *host*

The *service* program tried to lookup the name *host*, but was returned as an invalid reply.

313 *service*(*id*): Warning: user *user* from host *host* gave bad passcode

This message is generated only when Security Dynamics' ACE/Server security software is installed. The *user*'s password with was invalid according to the ACE server and access was denied.

314 Warning in user file (line *N*: null user name; ignoring)

While looking up a user name in the Gateway User Database, a bad entry with a blank user name was found.

315 Warning in user file (line *N*: uid not numeric; ignoring)

While looking up a user name in the Gateway User Database, a bad entry with a non-numeric user id was found.

316 Warning in user file (line *N*: gid not numeric; ignoring)

While looking up a user name in the Gateway User Database, a bad entry with a non-numeric group id was found.

317 Warning in group file (line *N*: null group name; ignoring)

While looking up a group in the Gateway User Database, a bad entry with a blank group name was found.

318 Warning in group file (line *N*: gid not numeric; ignoring)

While looking up a group in the Gateway User Database, a bad entry with a non-numeric group id was found.

323 Warning: readeagle already running

The `readeagle` program that communicates between **Eagle** and **Eaglet** was started up but it was already running.

331 Warning: no rules for '*eaglet*' in config file

On an **Eaglet**, it read the gateway configuration file but found no rules for itself.

333 Warning: can't open eaglets file /usr/adm/sg/eaglets

On an **Eagle**, the file exists but was unable to be opened for reading.

334 *service*(*id*): Warning: denied access to command '*comm and*'

FTP logs any attempt to use a restricted command. The option `ftp.getonly` stops all PUT and STORE commands, and the option `ftp.putonly` stops all GET and RETRIEVE commands. The command that was attempted is given.

B.4 Error Messages [severity Error]

Log messages in the range 400-499 are errors. Error messages are issued when normal gateway operation cannot complete successfully. The security of your gateway is still ensured, however you should attempt to correct any error that occurred. The message number, the message text, and a description of these messages follow:

401 Internal error: *error*

This message indicates a severe internal error in the **Eagle** software. Functionality has probably been degraded. Please report all such errors to Raptor Systems.

402 Can't get config file; Authorization machine may be down

`Gwcontrol` can't find its configuration file. This could mean that the Authorization machine may have crashed, or the private link between them broken.

403 Can't read config file `/usr/adm/sg/gateway.cf`

`Gwcontrol` couldn't open its config file for reading.

404 error in config file (line *N*: *error*)

While reading the config file, `gwcontrol` came across a serious error in the configuration file. The line number and error is reported. The entire line is ignored.

405 re-read of config file FAILED! Using previous config file

`Gwcontrol` received the command to reconfigure itself from its configuration file, but errors made it unable to do so. The old file is retained.

406 notifyd: cannot open audio file *file*

The audio file used in the `notifyd` configuration file `notify.cf` doesn't exist. This just means it can't ring the audio alarm.

407 traceroute: can't open lock file `/usr/adm/sg/traceroute`

The `traceroute` program cannot lock a file it needs to run.

408 notifyd: *file* is not a valid audio file In the Audio transport, the given audio file to play was not in Sun's u-law format.

409 notifyd: *file* sample rate *rate* not available In the Audio transport, the given audio file had a rate that the audio hardware can't play.

410 notifyd: *file* encoding not available: *type*

The audio *file* mentioned in the `notifyd` daemon's config file is not one it can use.

411 notifyd: unable to open notify schedule `/usr/adm/sg/notify.cf`

The `notifyd` daemon wasn't able to open its configuration file.

412 notifyd: unrecognized transport in `/usr/adm/sg/notify.cf` (*transport*)

The *transport* must be one of: *Mail, Pager, Audio, or Fax.*

413 Have config file errors, but continuing

`Gwcontrol`, when it has no previous configuration information to fall back on, will continue as well as possible.

415 notifyd: syntax error in date/time string: *string*

A date/time authorization string in the `notifyd` daemon's config file was badly formed. The point at which the error occurred is given in *string*.

416 notifyd: syntax error in expression string *string*

An expression string in the `notifyd` daemon's config file was badly formed. The point at which the error occurred is given in *string*.

417 passer: can't connect to *host* port *port* (*error*)

The `passer` program tried to connect to a server as given in its config file but failed. The reason is given in *error*.

418 passer: Line *N*: can't lookup host *host*

While reading its configuration file, `passer` failed looking up the *host*.

419 passer: Line *N*: bad server port *port*

While reading its configuration file, `passer` wasn't allowed to use a port.

420 passer: can't open config file `/usr/adm/sg/passer.cf`

`Passer` couldn't open its configuration file to read it.

421 passer: Line *N*: missing server host for *service*

The server host was left off on a line in `passer`'s config file.

422 passer: Line *N*: bad port number *port*

The *port* was not within the limits for port numbers.

423 passer: Line *N*: bad protocol *proto*

Proto was not UDP or TCP.

424 passer: Line *N*: can't use TCP port *port*

`Passer` is not allowed to use the FTP or Telnet ports.

425 passer: Line *N*: can't lookup service *service*

`Passer` couldn't look up the symbolic name *service* in the /etc/services file.

430 Can't lookup **Eagle** hostname *host*

An **Eaglet** couldn't look up the host name of its controlling **Eagle**.

431 Can't open `/usr/adm/sg/eagle.eyrie` (*error*)

An **Eaglet** couldn't read the location of its controlling **Eagle**.

432 Bad hostname in `/usr/adm/sg/eagle.eyrie`

An **Eaglet** couldn't read anything from the file that holds the name of its controlling **Eagle**.

433 Can't connect to *host* (*error*)

An attempt to contact the machine *host* failed.

434 Error reading config file

An error occurred while reading the **Eagle**'s configuration file.

435 Can't execute *program* to read config file (*error*)

An attempt to run the *program* failed.

440 readeagle-*eaglet*: Config load of **Eaglet**

The **Eagle** attempted to contact one of its **Eaglets**, but failed.

B.5 Alert Messages [severity Alert]

Log messages in the range 500-599 are alerts. Alert messages indicate a security rule has been triggered. This could mean that someone is attempting to violate your network. You should investigate when these messages are issued. The message number, the message text, and a description of these messages follow:

501 access from *incoming* to *outgoing* (*rule*): *period*

One of the security rules has been triggered. These rules indicate how many attempts from the hosts listed on the same line in the configuration file are allowed in a certain amount of time. By default, the security rule is 3 in 5 minutes, 5 in 15 minutes, 10 in 1 hour, 25 in one day, and 100 in one week.

502 ethernet address mismatch for *host*: ARP returns *etheraddr*

While checking the hardware ethernet address on *host*, a different one, *etheraddr*, than the one given in the config file was returned by ARP. The connection for *host* is dropped.

503 *service*(*id*): *host ipaddr1*: reverse address *ipaddr2* doesn't match

In trying to verify the identity of host *host*, the **Eagle** looked up the *host* name, and the returned *ipaddr2* didn't match the original *idaddr1*. This check is done to spot someone trying to spoof the *IP* address of another machine. The connection is dropped.

505 Unauthorized process killed (*process-id*), uid *user-id*

The `Vulture` routine found an unauthorized user program running and killed it.

B.6 Critical Messages [severity Critical]

Log messages in the range 600-699 are critical errors. These messages mean that the Eagle security is still in place, however an ancillary program has failed. Contact Raptor System when you see these messages. The message number, the message text, and a description of the messages follow:

601 Child process (*pid*) killed, signal *signal*

The **Eagle** software runs independent processes in the background to free `gwcontrol`. One of them apparently died unexpectedly.

602 Child process (*pid*) exited (*value*)

One of the **Eagle**'s software independent processes returned an unexpected error. There's probably another error message describing the problem better.

603 fork failed (*error*)

One of the **Eagle**'s software programs tried to fork, but failed. The Unix system has probably run out of swap space or processes. Reboot.

604 bad message priority (*num*)

The message priority must be one of: Debug, Info, Notice, Warning, Error, Alert, Critical, or Emergency. A message was generated with an invalid priority.

605 Can't execute *program* (*error*)

A program that should be there isn't there.

606 notifyd: failed to notify: transport=*transport*, priority=*severity*, *params*

The `notifyd` daemon failed to send a message using the *transport*.

607 Daemon exited on signal (*signal*)

One of the **Eagle**'s daemons exited due to a software signal.

B.7 Emergency Messages [severity Emergency]

Log messages in the range 700-799 indicate an emergency. This means that `gwcontrol` has failed and the system will no longer allow traffic to flow in or out of the G box. Security is ensured by shutting down all network traffic through your gateway. It is unlikely you will see these messages. If you do, investigate and try to correct the problem. If you cannot recover these errors, contact Raptor Systems for help. The message number, the message text, and a description of the messages follow:

701 FATAL: can't allocate memory

`Gwcontrol` has used up all its memory and cannot continue.

702 Quitting because of config errors

The syntax errors found in the configuration file are severe enough so that `gwcontrol` can't start running.

704 Expiration date reached (*date*); quitting

A demonstration copy of the **Eagle** software has reached its expiration date.

705 Only allowed to run on hostid *hostid*; quitting

Individual copies of the **Eagle** software are made so that they can be run only on a single machine. You probably installed from the wrong tape.

C DNS and Sendmail Configuration

C.1 DNS

There are several approaches to configuring the Domain Name Service (DNS) to work in conjunction with the Eagle. One is to run your primary DNS server directly on the G Box, with appropriate configuration of the *Generic Service Passer* file to allow DNS packets to enter and exit the internal network. This may not be appropriate if you wish to conceal the hostnames and IP addresses of your internal machines, or in the case where the IP addresses of your internal machines are not unique.

An alternative is to create a dual-level DNS system, using one or more special *root* nameservers on machines in your internal network. We will refer to these as internal root nameservers. In such a configuration, internal client machines query the internal root nameserver first, followed by the one on the G Box. The latter is capable of querying true root nameservers on the outside network. The G Box nameserver does not exchange DNS zone records for the internal network with outside nameservers.

In either case, it is assumed that you are providing primary name service for your domain, rather than you relying on your Internet service provider for primary name service. Your provider can function as a secondary, off-site nameserver, to meet Internet off-site nameserver requirements.

C.1.1 Dual-Level DNS

You can implement the above-described DNS configuration using the following steps.

C.1.2 Changes on the G Box

First, create a traditional DNS *cache* file (usually called `named.ca` in DNS documentation) on your G Box. Here is an example:

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; initial cache data for root domain servers
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
. 99999999 IN NS ns.nic.ddn.mil.
. 99999999 IN NS terp.umd.edu.
. 99999999 IN NS c.nyser.net.
. 99999999 IN NS kava.nisc.sri.com.
. 99999999 IN NS ns.nasa.gov.
. 99999999 IN NS aos.brl.mil.
;
ns.nic.ddn.mil. 99999999 IN A 192.112.36.4
terp.umd.edu. 99999999 IN A 128.8.10.90
c.nyser.net. 99999999 IN A 192.33.4.12
kava.nisc.src.com 99999999 IN A 192.33.33.24
ns.nasa.gov. 99999999 IN A 128.102.16.10
99999999 IN A 192.52.195.10
aos.brl.mil. 99999999 IN A 128.63.4.82
99999999 IN A 192.5.25.82
```

Your Internet service provider can give you an up-to-date DNS *cache* file, since the Internet root nameservers change from time to time. This file is consulted by the DNS system when it first starts (usually when your system is booted) and it contains permanent information about root nameservers accessible on the Internet.

Second, create the `named.boot` file. Here is an example:

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; named.boot file for G Box
;
primary your.domain db.domain
primary 987.654.321.in-addr.arpa db.123.456.789
primary 0.0.127.in-addr.arpa db.127.0.0
```

```
cache . named.cache
```

See your DNS documentation for the format of the files we've named *db.something* in this example.

Third, create DNS database files for your G Box (the *db.something* files noted above), but include in them only the information about your network and its hosts that you wish to be made generally available on the Internet via the Domain Name Service. Do not include any internal machines whose network identity you wish to keep internal to your network. *Most importantly, do not include any information about your internal root nameserver(s).* In this way, the nameserver running on the G Box will not be able to independently resolve hostnames on your internal network, except by querying the internal root nameserver. It not being able to do so also ensures that it cannot broadcast that information to the Internet at large. Tools to help you set up your DNS database files include `h2n`, a set of perl scripts which converts your /etc/hosts file to DNS database files (perl–Pratical Extration and Report Language–is a freely distributable package, available at most major anonymous ftp sites). `h2n` is available via anonymous ftp from `ftp.uu.net` in directory `/published/oreilly/nutshell/dnsbind`. HP-UX users can use the HP- specific tool `hosts_to_named` to accomplish the same objective.

Fourth, create the `resolv.conf` file on your G box to include references to both itself and the internal root nameserver(s). Here is an example:

```
######################################
#
# resolv.conf file for G Box
#
domain your.domain
nameserver 123.56.76.123 # IP address of internal root name-
server
nameserver 127.0.0.1 # this machine
```

This allows the G Box's nameserver to resolve hostnames by querying your internal nameserver(s) first, then itself. *Querying the internal nameserver first is a critical aspect of this setup.* This seems unusual, since the G Box is itself a nameserver, but since the G Box nameserver has limited information about your internal network, this configuration forces it to look first to the internal root nameserver for information and only later does it Internet (i.e., external) hostnames in the normal fashion by looking in its own cache and/or querying true Internet root nameservers.

C.1.3 Changes on the Internal Root Nameserver(s)

First, create a special version of the *cache* file for your internal root nameservers. This special file will not contain references to the Internet root nameservers like the one on your G Box. Rather, this file contains only a reference to the nameserver running on the G Box. Here is an example (be sure to substitute your G Box's hostname and IP address):

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; initial cache data for internal root domain servers
;
;
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
. 99999999 IN NS gbox.your.domain
gbox.your.domain 99999999 IN A 123.45.67.89
```

As we shall see below, the objective of using this abbreviated *cache* file on your internal root nameserver(s) is to keep it ignorant about the DNS system in the outside (Internet) world and to force it to query your G Box to resolve hostnames that it cannot otherwise resolve.

Second, create the `named.boot` file on your internal root nameserver(s) with the `forwarders` flag. Here is an example:

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; named.boot file for Internal root nameserver
;
primary your.domain db.domain
primary 987.654.321.in-addr.arpa db.123.456.789
primary 0.0.127.in-addr.arpa db.127.0.0
forwarders 123.45.67.89 ; IP address of G Box
cache . named.cache ; file with limited cache data
```

See your DNS documentation for the format of the files we've named *db-something* in this example.

Note here that the internal root nameserver is designated as the *primary* nameserver for your domain. You will recall that the `named.boot` file on the G Box also designates it (the G Box) as your domain's primary server. Both are correct in this configuration. To your internal hosts, the internal root nameserver is in fact primary for your domain. To the Internet at large, however, you have presented only a limited picture of your secured network, but have also presented the G Box nameserver as your primary nameserver. In other words, you are purposely presenting incomplete information about your network to the outside world, while keeping the complete information private with the internal root nameserver.

The third step in setting up your dual-level DNS system is to create all your other DNS database files (the *db.something* files noted above) on your internal root nameserver(s), as you normally would, listing all the hosts in your domain. Create these files in exactly the way your DNS documentation instructs you. Setting up these files enables your internal nameservers to resolve all internal hostnames without having to ask the nameserver on the G Box for help. Tools to help you set up your DNS database files include `h2n`, a set of perl scripts which converts your /etc/hosts file to DNS database files (perl–Pratical Extration and Report Language–is a freely distributable package, available at most major anonymous ftp sites). `h2n` is available via anonymous ftp from `ftp.uu.net` in directory `/published/oreilly/nutshell/dnsbind`. HP-UX users can use the HP- specific tool `hosts_to_named` to accomplish the same objective.

Last, create the `resolv.conf` file on your internal root nameservers), like this:

```
;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;
;
; resolv.conf file for the internal root nameserver
;
domain your.domain
nameserver 127.0.0.1 # this machine
nameserver 123.56.76.123 # IP address of other internal root
nameserver
nameserver 123.45.67.89 # IP address of G Box
```

This last step enables your internal nameserver(s) to query that on the G Box for names not locally resolvable (i.e., hostnames outside the internal, protected network). Note that this setup causes your internal root nameserver to query itself first, then any other internal root nameserver (if you have installed more than one), and, finally, the G Box.

As a final step, copy the `resolv.conf` file from your internal root nameserver to all of your internal clients, deleting the `127.0.0.1` line.

For more information on setting up *internal root nameservers*, see Chapter 14 of *DNS and BIND in a Nutshell,* written by Paul Albitz and Cricket Liu, published by O'Reilly and Associates, ISBN 1-56592-010-4.

One further note on DNS configuration. The Eagle software is compiled with the `resolv+` hostname lookup routines by Bill Wisner, which use `/etc/host.conf` to decide the order of name resolution. Accordingly, the Eagle-supplied versions of `telnet` and `ftp`, along with the internal hostname resolution software that is part of the Eagle, will follow the order of lookup specified in `/etc/host.conf`. If for some reason you want to change the default order of name resolution, create the `/etc/host.conf` file with your desired resolution sequence.

C.2 Sendmail

Because Sun's DNS routines are supported only when you are running NIS, and because the Eagle installation disables NIS on your system, several steps are necessary to configure the `sendmail` program. (These steps do not apply to AIX and HP-UX systems; just to Sun's.) Before doing so, obtain from your Internet service provider the hostname of its main mail-relay machine. The example below uses `relay.net.provider` so be sure to use your provider's true mail-relay hostname instead.

C.2.1 Configuring sendmail on SunOS

- Copy `/usr/lib/sendmail` to `/usr/lib/sendmail.nomx`.
- Copy `/usr/lib/sendmail.mx` to `/usr/lib/sendmail`.
- Remove (if necessary) `/etc/sendmail.fc`.
- Copy `/etc/sendmail.cf` to `/etc/sendmail.cf.old`.
- Copy `/usr/lib/sendmail.main.cf` to `/etc/sendmail.cf`.
- Edit the new `/etc/sendmail.cf` and make the following changes:
  1. Comment out the line `Dj$m` in the local info section near the top of the file (that is, change `Dj$m` to `#Dj$m`).
  2. Uncomment the line 4 lines below that says `#Dj$w` to make it `Dj$w`.
  3. Change `DMsmartuucp` to `DMddn`.
  4. Change `DRddn-gateway` to `DRrelay.net.provider` and `CRddn-gateway` to `CRrelay.net.provider`.

5. Add a new line just after the last two changes that is `Cw gboxname gbox-othername` (use the hostnames assigned to the G Box's two network interfaces).
6. Locate the two example lines which contain the word `Podunk` and use as a model to define your local domain name.

- Edit your /etc/rc.local script to specify your hostname as the fully qualified hostname (i.e., add a line, just before the line which begins with `ifconfig` which says `hostname hostname.your.domain` substituting the full hostname of your system, with your Internet domain name appended).
- Reboot your system
- Test out the reconfigured mail with:

```
/usr/lib/sendmail -v < /dev/null e-mail_address@some.host.outside
```

  This runs sendmail in *verbose* mode, and you can watch the name resolution and SMTP conversation in progress; it sends an empty mail message to the address specified.

C.2.2 Configuring sendmail under AIX

The `/etc/sendmail.cf` file on IBM RS/6000 systems needs editing only to specify the name of the Internet relay system (see the `relay.net.provider` instructions above) and to add the `Cw` line, as described above.

C.2.3 Configuring sendmail under HP-UX

The `/usr/lib/sendmail.cf` file on HP-UX systems needs editing only to specify the name of the Internet relay system (see the `relay.net.provider` instructions above) and to add the `Cw` line, as described above. In addition, edit the `/etc/netbsdsrc` file and comment out the line which creates the frozen sendmail configuration (`/usr/lib/sendmail -bz`). Remove any `/etc/sendmail.fc` file before rebooting.

C.2.4 Futher E-Mail Configuration

To allow bi-directional e-mail access for your internal machines (that is, to allow an internal machine to exchange e-mail with external hosts), further steps are necessary. The first step is to use a network-wide `/etc/aliases` file on the G Box (or your main mail host) to forward mail to users on internal machines. All e-mail to users on internal machines should be addressed to the G Box or mailhost. There follows an extract from a hypothetical `/etc/aliases` file which illustrates this approach. It assumes a small network consisting of the Eagle, a central mail-handling host named `mailhub`, two other hosts named `hawk` and `falcon`, and five users, `bob`, `mike`, `ellen`, `mary`, and `chris`. Of these users, the first two receive their e-mail on the mailhub and the others on individual hosts in the network.

```
bob:  bob@mailhub
mike:  mike@mailhub
ellen:  ellen@hawk
mary:  mary@falcon
chris:  chris@falcon
```

Using this `/etc/aliases` file, mail to all of the aliased users from the outside world would be addressed to them on the Eagle Gateway, and *not to the internal machines*. Thus, mail to mary would be addressed to `mary@gbox`, not `mary@falcon`. When mail to mary arrives, it will automatically be forwarded to her mailox on falcon.

To further simplify this approach, enable outside users to mail directly to your domain name using the `username@domain.name` syntax; do this by publishing a DNS mail-exchanger (MX) record for the domain itself using the same IP address as your G Box, as well as a normal DNS Address (A) record, again using the same IP address.

You can also modify the `/etc/sendmail.cf` file on your internal machines to rewrite the `From` line of outgoing messages to show the Eagle's hostname in place of the internal hostname. Making this change will cause all outgoing e-mail from machine in your network to appear to have come directly from the Eagle and all replies to such messages will be sent directly to the Eagle, then forwarded as specified in the `/etc/aliases` file you have set up. This is illustrated in the tutorial section of the book *sendmail*, by Bryan Costales, with Eric Allman and Neil Rickert, published by O'Reilly & Associates, ISBN 1-56592-056-2. A sample `client.cf` file which implements this procedure is available via anonymous ftp from the host `ftp.uu.net` in the directory `/published/oreilly/nutshell/sendmail`. If you obtain this file, be sure to edit it to change the name of the *mailhub* to your local mail hub system and to include your domain name.

Other steps you may want to take include creating MX records for each of your internal machines with your G Box as the mail exchanger.

For more information on configuring e-mail, including the topics outlined above, see the *sendmail* and *DNS and BIND* books.

D Questions And Answers

Below is a list of questions with answers that should give you better insight into how the Eagle helps to protect your network.

1. Why are two separate machines required for the Eagle?

   The security policy for your network is defined in the authorization rule database. This database resides on a separate machine that can be physically secured, preventing unauthorized changes. The rules are read by the Eagle on the gateway machine through a read-only, private connection, unreachable from anywhere in the network. Only those who have physical access to the authorization machine can modify the authorization rule database. No one can remotely modify your rule database–not even you, unless you have left some security hole open, such as an inappropriate `/.rhosts` file.

2. Can the Eagle break in a way which will compromise my security?

   Each packet must be specifically authorized by the Eagle as defined by your security policy before it is forwarded to your internal network or from your internal network to the outside. In the unlikely event Eagle terminates, all traffic through the gateway ceases. This results in loss of service rather than compromising security.

   Eagle counters the possible threat of modification of its code by detecting any code changes and immediately terminating. Again, this would cause loss of service through your gateway, preventing security breaches.

3. My LAN is Token Ring (or FDDI). Will Eagle work with this LAN?

   Yes. Your system's network adapter card (Token Ring or FDDI) must support the IP protocols to run the Eagle software.

4. Does Eagle create a bottleneck on my LAN?

   Forcing all transmissions to pass through the Eagle might appear to pose a performance problem. In practice, however, the performance is bounded by the network connection between your inside and outside network. To provide the highest possible security all network traffic is approved or denied by the Eagle but only adds a few milliseconds to each operation through the gateway. Network traffic on your inside network is unaffected. Only traffic to outside services is considered by the Eagle.

5. What is the throughput performance of Eagle?

   An *out of the box* 25MHz sparc processor can pass ethernet based TCP/IP packets at approximately 250KB/sec. With a high speed line interface T1 speeds can be closely approached. Faster machines can, of course, pass data more quickly.

6. Why can't I have user accounts on the Eagle gateway machine?

   Since all operating systems have known and unknown security weaknesses, a person logged into the machine might discover and exploit these weaknesses. Eagle allows root login only. Any non-rooted program is terminated automatically by the Vulture program.

7. What are the known weaknesses of TCP/IP?

   The biggest problem with TCP/IP is that you must assume that the information in an incoming packet header is valid. This may not be the case; a host may assume the IP address of another machine. Eagle cannot directly detect this *spoofing*, except on your local network with Etherguard. Other features of the Eagle however, such as the Dynamic Activity Monitor tend to minimize most outside attacks. Opportunities for spoofing are further reduced by restricting sites by service (i.e. `ftp` only) or to a particular time of day (i.e. restrict sites to day-time access only since many attacks occur at night).

   Eagle gives additional protection by preventing outside machines from spoofing an address from an internal machine on your network. Even routers, which are sometimes used to protect networks, cannot provide for this type of attack. For more information on network security matters, see the publications listed in the preface.

8. Is there a *back door* to the Eagle?

   No. No one can access the Eagle remotely, not even *Raptor Systems*. Eagle was designed to allow access from the console of the gateway machine only.

9. We have a large number of systems on our network, but don't have a NIC-issued IP address family. Instead, we have used an arbitrary set of IP addresses that are probably already used on the Internet. We'd rather not have to renumber our entire network. Can The Eagle operate in this environment?

   Many of our customers are in a similar situation. Eagle hides the internal structure of your network, including its IP addresses. All outgoing IP packets are rewritten at our gateway to replace the internal network information with that of the Gateway, with the result that to the outside world, it appears that all communications with your network come to/from the Eagle gateway alone.

10. We have a NIC-issued IP address family, but want to maintain confidentiality of the hostnames/IP addresses of systems inside our network. How can we manage this without broadcasting all the information about our internal network with DNS?

    This, too, is a common need. The Eagle works in this situation as well, using the same packet-rewrite mechanism.

    In both of these cases, customers can take advantage of the Internet Domain Name Service (DNS) even though the IP addresses of their internal hosts are never made available to the Internet at large. We recommend a two-level DNS arrangement for our customers, using what are known as *internal root nameservers* (also called *false* root nameservers) to preserve internal network confidentiality or deal with the problem of duplicate IP addresses. This setup allows internal hosts to be able to resolve Internet hostnames/addresses within the Eagle-secured nework, but prevents resolution of internal hostnames by outside hosts, effectively dealing with both of these issues. DNS information about your internal network is not broadcast to the outside world and, since all packets are rewritten at the gateway, the outside system knows only that it is communicating with the Eagle gateway instead of internal systems. Re-writing outgoing e-mail headers (as described below) completes the picture of keeping information about your internal network confidential.

11. Can the Eagle enable us to present a single interface to the Internet for electronic mail addresses for all our internal users? What about forcing all Internet e-mail to come into a specific mail gateway system?

    The first of these goals can be achieved in an Eagle-secured network, although there is nothing special about the Eagle software itself which enables it. A combination of setup of e-mail aliases, DNS mail-exchanger (MX) records, and reconfiguration of sendmail on your UNIX hosts inside the network and on the Eagle gateway system itself will enable addressing of e-mail to names like `yourname@yourcompany.com` and ensure that outgoing e-mail contains such a return address, regardless of where it arises inside your network.

Raptor Systems can make available software which will, in combination with the Eagle software, automatically redirect all incoming e-mail addressed to any system in your network to the e-mail *hub* you specify. SMTP (e-mail) network packets coming in from the Internet will not be allowed to reach any system inside your network other than the specified mail hub, regardless of the actual address on the message. You can then turn off the sendmail transport agent on your gateway (and on all your internal hosts) altogether.

12. What Sort of Activity Reports Does the Eagle Provide?

    These can be generated from the log files by whatever techniques the customer chooses. The logs are flat files which can be easily parsed with tools ranging from simple UNIX shell utilities (such as sed, awk, cut, paste, etc.) to sophisticated database systems with full query language and reporting capabilities. The log files can be automatically emailed from the gateway on a daily basis to the analysis system, including being piped directly to data-analysis scripts for immediate processing.

13. Can the Eagle Allow Access to the World-Wide Web?

    Yes, via a proxy web server outside the firewall and the Generic Service Passer. You should be aware that use of a WWW proxy service can negate service limits you place in the Eagle authorization file, since WWW clients have ftp, telnet, and other services built in which would not be controllable with the Eagle software and which may require you to reconsider your security policy. It makes little sense to attempt to control outgoing ftp, for example, with the Eagle authorization rules, then let users access ftp services via their WWW browser and the proxy mechanism. Further, since the proxy WWW server operates *outside* the Eagle firewall, the server itself should not be considered secure.

E Sample Fax Output

Raptor Systems Incorporated

Eagle
Secure
Gateway

To: Kilgore Trout

Date: Thu Apr 2 16:47:32 1992

Hostname: gatekeeper

Priority: Alert

Message:

501 Access from cracker.edu to mymachine [rule line 2]: over 3 tries in 5 minutes.

Raptor Systems Incorporated
69 Hickory Drive, Waltham, Massachusetts 02254