Cisco − Improving Security on Cisco Routers

Table of Contents

Improving Security on Cisco Routers

Interactive: This document offers customized analysis of your Cisco device.

Introduction

This document is an informal discussion of some Cisco configuration settings that network administrators should consider changing on their routers, especially on their border routers, in order to improve security. This document is about basic, "boilerplate" configuration items that are almost universally applicable in IP networks, and about a few unexpected items of which you should be aware.

If you have the output of a **show running−configuration** command from your Cisco device, you can use to display potential issues and fixes. To use, you must be a registered customer, be logged in, and have JavaScript enabled.

You can use Output Interpreter to display potential issues and fixes. To use Output Interpreter, you must be a registered customer, be logged in, and have JavaScript enabled.

Before You Begin

Conventions

For more information on document conventions, see the Cisco Technical Tips Conventions.

Prerequisites

There are no specific prerequisites for this document.

Components Used

This document is not restricted to specific software and hardware versions.

Background Information

This is not an exhaustive list, nor can it be substituted for understanding on the part of the network administrator; it's simply a reminder of some of the things that are sometimes forgotten. Only commands that are important in IP networks are mentioned. Many of the services that can be enabled in Cisco routers require careful security configuration, but this document concerns itself mainly with services that are enabled by default, or that are almost always enabled by users, and that may need to be disabled or reconfigured.

This is particularly important because some of the default settings in Cisco IOS software are there for historical reasons; they made sense when they were chosen, but would probably be different if new defaults were chosen today. Other defaults make sense for most systems, but may create security exposures if they're used in devices that form part of a network perimeter defense. Still other defaults are actually required by standards, but aren't always desirable from a security point of view.

Cisco IOS software has many security−specific features, such as packet−filtering access lists, the Cisco IOS Firewall Feature Set, TCP Intercept, AAA, and encryption. Many other features, such as packet logging and quality of service features, can be used to increase network security against various attacks. None of these are discussed, except in passing. This is not a document about firewall configuration; for the most part, it is a document about securing the router itself, and ignores the equally important issue of protecting other network devices.

Password Management

Passwords (and similar secrets, such as SNMP community strings) are the primary defense against unauthorized access to your router. The best way to handle most passwords is to maintain them on a TACACS+ or RADIUS authentication server. However, almost every router will still have a locally configured password for privileged access, and may also have other password information in its configuration file.

enable secret

The enable secret command is used to set the password that grants privileged administrative access to the IOS system. An enable secret password should always be set. You should use **enable secret**, *not* the older **enable password**. **enable password** uses a weak encryption algorithm (see the description of the "service password−encryption" command).

If no enable secret is set, and a password is configured for the console TTY line, the console password may be used to get privileged access, even from a remote VTY session. This is almost certainly not what you want, and is another reason to be certain to configure an enable secret.

service password−encryption (and its limitations)

The **service password−encryption** command directs the IOS software to encrypt the passwords, CHAP secrets, and similar data that are saved in its configuration file. This is useful for preventing casual observers from reading passwords, for example, when they happen to look at the screen over an administrator's shoulder.

However, the algorithm used by **service password−encryption** is a simple Vigenere cipher; any competent amateur cryptographer could easily reverse it in at most a few hours. The algorithm was not designed to protect configuration files against serious analysis by even slightly sophisticated attackers, and should not be used for this purpose. Any Cisco configuration file that contains encrypted passwords should be treated with the same care used for a cleartext list of those same passwords.

This weak encryption warning does not apply to passwords set with the **enable secret** command, but it does apply to passwords set with **enable password**.

The **enable secret** command uses MD5 for password hashing. The algorithm has had considerable public review, and is not reversible as far as anybody at Cisco knows. It is, however, subject to dictionary attacks (a "dictionary attack" is having a computer try every word in a dictionary or other list of candidate passwords). It's therefore wise to keep your configuration file out of the hands of untrusted people, especially if you're not sure your passwords are well chosen.

Controlling Interactive Access

Anyone who can log in to a Cisco router can display information which you probably don't want to make available to the general public. A user who can log in to the router may be able to use it as a relay for further network attacks. Anyone who can get privileged access to the router can reconfigure it. To prevent inappropriate access, you need to control interactive logins to the router.

Although most interactive access is disabled by default, there are exceptions; the most obvious being interactive sessions from directly connected asynchronous terminals, such as the console terminal, and from integrated modem lines.

Console Ports

It's important to remember that the console port of an IOS device has special privileges. In particular, if a BREAK signal is sent to the console port during the first few seconds after a reboot, the password recovery procedure can easily be used to take control of the system. This means that attackers who can interrupt power or induce a system crash, and who have access to the console port via a hardwired terminal, a modem, a terminal server, or some other network device, can take control of the system, even if they do not have

physical access to it or the ability to log in to it normally.

It follows that any modem or network device that gives access to the Cisco console port must itself be secured to a standard comparable to the security used for privileged access to the router. At a bare minimum, any console modem should be of a type that can require the dialup user to supply a password for access, and the modem password should be carefully managed.

General Interactive Access

There are more ways of getting interactive connections to routers than users may realize. Cisco IOS software, depending on the configuration and software version, may support connections via Telnet; rlogin; SSH; non−IP−based network protocols like LAT, MOP, X.29, and V.120, and possibly other protocols; as well as via local asynchronous connections and modem dial−ins. More protocols for interactive access are always being added. Interactive Telnet access is available not only on the standard Telnet TCP port (port 23), but on a variety of higher−numbered ports as well.

All interactive access mechanisms use the IOS TTY abstraction (in other words, they all involve sessions on "lines" of one sort or another). Local asynchronous terminals and dialup modems use standard lines, known as "TTYs". Remote network connections, regardless of the protocol, use virtual TTYs, or "VTYs". The best way to protect a system is to make certain that appropriate controls are applied on all lines, including both VTY lines and TTY lines.

Because it's difficult to be certain that all possible modes of access have been blocked, administrators should usually make sure that logins on all lines are controlled using some sort of authentication mechanism, even on machines that are supposed to be inaccessible from untrusted networks. This is especially important for VTY lines and for lines connected to modems or other remote access devices.

Interactive logins may be completely prevented on any line by configuring it with the **login** and **no password** commands. This is the default configuration for VTYs, but not for TTYs. There are many ways to configure passwords and other forms of user authentication for TTY and VTY lines; consult the Cisco IOS software documentation for more information.

Controlling TTYs

Local asynchronous terminals are less common than they once were, but they still exist in some installations. Unless the terminals are physically secured, and usually even if they are, the router should be configured to require users on local asynchronous terminals to log in before using the system. Most TTY ports in modern routers are either connected to external modems, or are implemented by integrated modems; securing these ports is obviously even more important than securing local terminal ports.

By default, a remote user can establish a connection to a TTY line over the network; this is known as "reverse Telnet," and allows the remote user to interact with the terminal or modem connected to the TTY line. It's possible to apply password protection for such connections. Often, it's desirable to allow users to make connections to modem lines, so that they can make outgoing calls. However, this feature may allow a remote user to connect to a local asynchronous terminal port, or even to a dial−in modem port, and simulate the router's login prompt to steal passwords, or to do other things that may trick local users or interfere with their work.

To disable this reverse Telnet feature, apply the configuration command **transport input none** to any asynchronous or modem line that shouldn't be receiving connections from network users. If at all possible, don't use the same modems for both dial−in and dial−out, and don't allow reverse Telnet connections to the lines you use for dial−in.

Controlling VTYs and Ensuring VTY Availability

Any VTY should be configured to accept connections only with the protocols actually needed. This is done with the **transport input** command. For example, a VTY that was expected to receive only Telnet sessions would be configured with **transport input telnet**, while a VTY permitting both Telnet and SSH sessions would have **transport input telnet ssh**. If your software supports an encrypted access protocol such as SSH, it may be wise to enable only that protocol, and to disable cleartext Telnet. It's also usually a good idea to use the **ip access−class** command to restrict the IP addresses from which the VTY will accept connections.

A Cisco IOS device has a limited number of VTY lines (usually five). When all of the VTYs are in use, no more remote interactive connections can be established. This creates the opportunity for a denial−of−service attack; if an attacker can open remote sessions to all the VTYs on the system, the legitimate administrator may not be able to log in. The attacker doesn't have to log in to do this; the sessions can simply be left at the login prompt.

One way of reducing this exposure is to configure a more restrictive **ip access−class** command on the last VTY in the system than on the other VTYs. The last VTY (usually VTY 4) might be restricted to accept connections only from a single, specific administrative workstation, whereas the other VTYs might accept connections from any address in a corporate network.

Another useful tactic is to configure VTY timeouts using the **exec−timeout** command. This prevents an idle session from consuming a VTY indefinitely. Although its effectiveness against deliberate attacks is relatively limited, it also provides some protection against sessions accidentally left idle. Similarly, enabling TCP keepalives on incoming connections (with **service tcp−keepalives−in**) can help to guard against both malicious attacks and "orphaned" sessions caused by remote system crashes.

Complete VTY protection can be provided by disabling all non−IP−based remote access protocols, and using IPSec encryption for all remote interactive connections to the router. IPSec is an extra−cost option, and its configuration is beyond the scope of this document.

Warning Banners

In some jurisdictions, civil and/or criminal prosecution of crackers who break into your systems is made much easier if you provide a banner informing unauthorized users that their use is in fact unauthorized. In other jurisdictions, you may be forbidden to monitor the activities of even unauthorized users unless you have taken steps to notify them of your intent to do so. One way of providing this notification is to put it into a banner message configured with the Cisco IOS **banner login** command.

Legal notification requirements are complex, and vary in each jurisdiction and situation. Even within jurisdictions, legal opinions vary, and this issue should be discussed with your own legal counsel. In cooperation with counsel, you should consider which of the following information should be put into your banner:

- A notice that the system is to be logged in to or used only by specifically authorized personnel, and perhaps information about who may authorize use.
- A notice that any unauthorized use of the system is unlawful, and may be subject to civil and/or criminal penalties.
- A notice that any use of the system may be logged or monitored without further notice, and that the resulting logs may be used as evidence in court.
- Specific notices required by specific local laws.

From a security, rather than a legal, point of view, your login banner usually should not contain any specific information about your router, its name, its model, what software it's running, or who owns it; such information may be abused by crackers.

Commonly Configured Management Services

Many users manage their networks using protocols other than interactive remote login. The most common protocols for this purpose are SNMP and HTTP.

Neither of these protocols is enabled by default, and, as for any other service, the most secure option is not to enable them at all. However, if they are enabled, they should be secured as described here.

SNMP

SNMP is very widely used for router monitoring, and frequently for router configuration changes as well. Unfortunately, version 1 of the SNMP protocol, which is the most commonly used, uses a very weak authentication scheme based on a "community string," which amounts to a fixed password transmitted over the network without encryption. If at all possible, use SNMP version 2, which supports an MD5−based digest authentication scheme, and allows for restricted access to various management data.

If you must use SNMP version 1, you should be careful to choose inobvious community strings (not, for example, "public" or "private"). If at all possible, you should avoid using the same community strings for all network devices; use a different string or strings for each device, or at least for each area of the network. Do not make a read−only string the same as a read−write string. If possible, periodic SNMP version 1 polling should be done with a read−only community string; read−write strings should be used only for actual write operations.

SNMP version 1 is ill suited to use across the public Internet for the following reasons:

- It uses cleartext authentication strings.
- Most SNMP implementations send those strings repeatedly as part of periodic polling.
- It is an easily spoofable, datagram−based transaction protocol.

You should carefully consider the implications before using it that way.

In most networks, legitimate SNMP messages will come only from certain management stations. If this is true in your network, you should probably use the access list number option on the **snmp−server community** command to restrict SNMP version 1 access to only the IP addresses of the management stations. Do not use the **snmp−server community** command for any purpose in a pure SNMP version 2 environment; this command implicitly enables SNMP version 1.

For SNMP version 2, configure digest authentication with the **authentication** and **md5** keywords of the **snmp−server party** configuration command. If possible, use a different MD5 secret value for each router.

SNMP management stations often have large databases of authentication information, such as community strings. This information may provide access to many routers and other network devices. This concentration of information makes the SNMP management station a natural target for attack, and it should be secured accordingly.

HTTP

Most recent Cisco IOS software versions support remote configuration and monitoring using the World Wide Web's HTTP protocol. In general, HTTP access is equivalent to interactive access to the router. The authentication protocol used for HTTP is equivalent to sending a cleartext password across the network, and, unfortunately, there is no effective provision in HTTP for challenge−based or one−time passwords. This makes HTTP a relatively risky choice for use across the public Internet.

If you choose to use HTTP for management, you should restrict access to appropriate IP addresses using the **ip http access−class** command. You should also configure authentication using the **ip http authentication** command. As with interactive logins, the best choice for HTTP authentication is probably to use a TACACS+ or RADIUS server. It's usually wisest to avoid using the "enable" password as an HTTP password.

Management and Interactive Access via the Internet (and Other Untrusted Networks)

Many users manage their routers remotely, and sometimes this is done over the Internet. Any unencrypted remote access carries some risk, but access over a public network such as the Internet is especially dangerous. All remote management schemes, including interactive access, HTTP, and SNMP, are vulnerable.

The attacks discussed in this section are relatively sophisticated ones, but they are by no means out of the reach of today's crackers. These attacks can often be thwarted if the public network providers involved have taken proper security measures; you will need to evaluate your level of trust in the security measures used by all the providers carrying your management traffic. Even if you trust your providers, it's usually wise to take at least some steps to protect yourself from the results of any mistakes they may make.

All the cautions here apply as much to hosts as to routers. This document talks about protecting router login sessions, but you should look into using analogous mechanisms to protect your hosts if you administer those hosts remotely.

Remote Internet administration is useful, but requires careful attention to security.

Packet Sniffers

Crackers frequently break into computers owned by Internet service providers, or into computers on other large networks, and install "packet sniffer" programs, which monitor traffic passing through the network and steal data such as passwords and SNMP community strings. Although this is becoming more difficult as network operators improve their security, it's still relatively common. In addition to the risk from outside crackers, it's not unheard of for rogue ISP personnel to install sniffers. Any password sent over an unencrypted channel is at risk, and this includes the login and enable passwords for your routers.

If at all possible, you should avoid logging in to your router using any unencrypted protocol over any untrusted network. If your router software supports it, it's a good idea to use an encrypted login protocol such as SSH or Kerberized Telnet. Another possibility is to use IPSec encryption for all router management traffic, including Telnet, SNMP, and HTTP. All of these encryption features are subject to certain export restrictions imposed by the United States Government, and are special−order, extra−cost items on Cisco routers.

If you don't have access to an encrypted remote access protocol, another possibility is to use a one−time password system such as S/KEY or OPIE, together with a TACACS+ or RADIUS server, to control both interactive logins and privileged access to your router. The advantage here is that a stolen password is of no use, since it is made invalid by the very session in which it is stolen. Non−password data transmitted in the

session remain available to eavesdroppers, but many sniffer programs are set up to concentrate on passwords.

If you absolutely must send passwords over cleartext Telnet sessions, you should change your passwords frequently, and pay close attention to the path traversed by your sessions.

Other Internet Access Dangers

In addition to packet sniffers, remote Internet management of routers presents the following security risks:

- In order to manage a router over the Internet, you must permit at least some Internet hosts to have access to the router. It's possible that these hosts could be compromised, or that their addresses could be spoofed. By permitting interactive access from the Internet, you make your security dependent not only on your own anti−spoofing measures, but on those of the service providers involved.

  These dangers can be reduced by making sure that all the hosts that are permitted to log into your router are under your own control, and by using encrypted login protocols with strong authentication.
- It's sometimes possible to "hijack" an unencrypted TCP connection (such as a Telnet session), and actually take control away from a user who's logged in. Although such hijacking attacks aren't nearly as common as simple packet sniffing, and although they can be complex to mount, they are possible, and might be used by an attacker who had your network specifically in mind as a target. The only real solution to the problem of session hijacking is to use a strongly authenticated encrypted management protocol.
- Denial of service attacks are relatively common on the Internet. If your network is being subjected to a denial of service attack, you may not be able to reach your router to collect information or take defensive action. Even an attack on someone else's network may impair your management access to your own network. Although you can take steps to make your network more resistant to denial of service attacks, the only real defense against this risk is to have a separate, out−of−band management channel, such as a dialup modem, for use in emergencies.

Logging

Cisco routers can record information about a variety of events, many of which have security significance. Logs can be invaluable in characterizing and responding to security incidents. The main types of logging used by Cisco routers are:

- AAA logging, which collects information about user dial−in connections, logins, logouts, HTTP accesses, privilege level changes, commands executed, and similar events. AAA log entries are sent to authentication servers using the TACACS+ and/or RADIUS protocols, and are recorded locally by those servers, typically in disk files. If you are using a TACACS+ or RADIUS server, you may wish to enable AAA logging of various sorts; this is done using AAA configuration commands such as **aaa accounting**. Detailed description AAA configuration is beyond the scope of this document.
- SNMP trap logging, which sends notifications of significant changes in system status to SNMP management stations. You will probably want to use SNMP traps only if you have a preexisting SNMP management infrastructure.
- System logging, which records a large variety of events, depending on the system configuration. System logging events may be reported to a variety of destinations, including the following:

  - The system console port (**logging console**).
  - Servers using the UNIX "syslog" protocol (**logging *ip−address*, logging trap**).
  - Remote sessions on VTYs and local sessions on TTYs (**logging monitor, terminal monitor**).
  - A local logging buffer in router RAM (**logging buffered**).

From a security point of view, the most important events usually recorded by system logging are interface status changes, changes to the system configuration, access list matches, and events detected by the optional firewall and intrusion detection features.

Each system logging event is tagged with an urgency level. The levels range from debugging information (at the lowest urgency), to major system emergencies. Each logging destination may be configured with a "threshold" urgency, and will receive logging events only at or above that threshold.

Saving Log Information

By default, system logging information is sent only to the asynchronous console port. Since many console ports are unmonitored, or are connected to terminals without historical memory and with relatively small displays, this information may not be available when it's needed, especially when a problem is being debugged over the network.

Almost every router should save system logging information to a local RAM buffer. The logging buffer is of a fixed size, and retains only the newest information. The contents of the buffer are lost whenever the router is reloaded. Even so, even a moderately−sized logging buffer is often of great value. On low−end routers, a reasonable buffer size might be 16384 or 32768 bytes; on high−end routers with lots of memory (and many logged events), even 262144 bytes might be appropriate. You can use the **show memory** command to make sure that your router has enough free memory to support a logging buffer. Create the buffer using the **logging buffered *buffer−size*** configuration command.

Most larger installations will have "syslog" servers. You can send **logging** information to a server with logging server−ip−address, and you can control the urgency threshold for logging to the server with **logging trap *urgency*** . Even if you have a syslog server, you should probably still enable local logging.

If your router has a real−time clock or is running NTP, you will probably want to time−stamp log entries using **service timestamps log datetime msecs**.

Recording Access List Violations

If you use access lists to filter traffic, you may want to log packets that violate your filtering criteria. Older Cisco IOS software versions support logging using the **log** keyword, which causes logging of the IP addresses and port numbers associated with packets matching an access list entry. Newer versions provide the **log−input** keyword, which adds information about the interface from which the packet was received, and the MAC address of the host that sent it.

It's not usually a good idea to configure logging for access list entries that will match very large numbers of packets. Doing so will cause log files to grow excessively large, and may cut into system performance. However, access list log messages are rate−limited, so the impact is not catastrophic.

Access list logging can also be used to characterize traffic associated with network attacks, by logging the suspect traffic.

Securing IP Routing

This section discusses some basic security measures related to the way in which the router forwards IP packets. More information about these issues is available in this document about essential IOS features.

Anti−spoofing

Many network attacks rely on an attacker falsifying, or "spoofing," the source addresses of IP datagrams. Some attacks rely on spoofing to work at all, and other attacks are much harder to trace if the attacker can use somebody else's address instead of his or her own. Therefore, it's valuable for network administrators to prevent spoofing wherever feasible.

Anti−spoofing should be done at every point in the network where it's practical, but is usually both easiest and most effective at the borders between large address blocks, or between domains of network administration. It's usually impractical to do anti−spoofing on every router in a network, because of the difficulty of determining which source addresses may legitimately appear on any given interface.

If you're an Internet service provider (ISP), you may find that effective anti−spoofing, together with other effective security measures, causes expensive, annoying problem subscribers to take their business to other providers. ISPs should be especially careful to apply anti−spoofing controls at dialup pools and other end−user connection points (see also RFC 2267).

Administrators of corporate firewalls or perimeter routers sometimes install anti−spoofing measures to prevent hosts on the Internet from assuming the addresses of internal hosts, but don't take steps to prevent internal hosts from assuming the addresses of hosts on the Internet. It's a far better idea to try to prevent spoofing in both directions. There are at least three good reasons for doing anti−spoofing in both directions at an organizational firewall:

1. Internal users will be less tempted to try launching network attacks and less likely to succeed if they do try.
2. Accidentally misconfigured internal hosts will be less likely to cause trouble for remote sites (and therefore less likely to generate angry telephone calls or damage your organization's reputation).
3. Outside crackers often break into networks as launching pads for further attacks. These crackers may be less interested in a network with outgoing spoofing protection.

Anti−spoofing with access lists

Unfortunately, it's not practical to give a simple list of commands that will provide appropriate spoofing protection; access list configuration depends too much on the individual network. However, the basic goal is simple: to discard packets that arrive on interfaces that are not viable paths from the supposed source addresses of those packets. For example, on a two−interface router connecting a corporate network to the Internet, any datagram that arrives on the Internet interface, but whose source address field claims that it came from a machine on the corporate network, should be discarded.

Similarly, any datagram arriving on the interface connected to the corporate network, but whose source address field claims that it came from a machine outside the corporate network, should be discarded. If CPU resources allow it, anti−spoofing should be applied on any interface where it's feasible to determine what traffic may legitimately arrive.

ISPs carrying transit traffic may have limited opportunities to configure anti−spoofing access lists, but such an ISP can usually at least filter outside traffic that claims to originate within the ISP's own address space.

In general, anti−spoofing filters must be built with input access lists; that is, packets must be filtered at the interfaces through which they arrive at the router, not at the interfaces through which they leave the router. This is configured with the **ip access−group *list* in** interface configuration command. It's possible to do anti−spoofing using output access lists in some two−port configurations, but input lists are usually easier to understand even in those cases. Furthermore, an input list protects the router itself from spoofing attacks,

whereas an output list protects only devices "behind" the router.

When anti−spoofing access lists exist, they should always reject datagrams with broadcast or multicast source addresses, and datagrams with the reserved "loopback" address as a source address. It's usually also appropriate for an anti−spoofing access list to filter out all ICMP redirects, regardless of source or destination address. Appropriate commands would be:

```
access−list number deny icmp any any redirect
access−list number deny ip 127.0.0.0 0.255.255.255 any
access−list number deny ip 224.0.0.0 31.255.255.255 any
access−list number deny ip host 0.0.0.0 any
```

Note that the fourth command will filter out packets from many BOOTP/DHCP clients, and therefore is not appropriate in all environments.

Anti−spoofing with RPF checks

In almost all Cisco IOS software versions that support Cisco Express Forwarding (CEF), it's possible to have the router check the source address of any packet against the interface through which the packet entered the router. If the input interface isn't a feasible path to the source address according to the routing table, the packet will be dropped.

This works only when routing is symmetric. If the network is designed in such a way that traffic from host A to host B may normally take a different path than traffic from host B to host A, the check will always fail and communication between the two hosts will be impossible. This sort of asymmetric routing is common in the Internet core. You should make sure that your network doesn't use asymmetric routing before enabling this feature.

This feature is known as a reverse path forwarding (RPF) check, and is enabled with the command **ip verify unicast rpf**. It is available in Cisco IOS software 11.1CC, 11.1CT, 11.2GS, and all 12.0 and later versions, but requires that CEF be enabled in order to be effective.

Controlling Directed Broadcasts

IP directed broadcasts are used in the extremely common and popular "smurf" denial of service attack, and can also be used in related attacks.

An IP directed broadcast is a datagram which is sent to the broadcast address of a subnet to which the sending machine is not directly attached. The directed broadcast is routed through the network as a unicast packet until it arrives at the target subnet, where it is converted into a link−layer broadcast. Because of the nature of the IP addressing architecture, only the last router in the chain, the one that is connected directly to the target subnet, can conclusively identify a directed broadcast. Directed broadcasts are occasionally used for legitimate purposes, but such use is not common outside the financial services industry.

In a "smurf" attack, the attacker sends ICMP echo requests from a falsified source address to a directed broadcast address, causing all the hosts on the target subnet to send replies to the falsified source. By sending a continuous stream of such requests, the attacker can create a much larger stream of replies, which can completely inundate the host whose address is being falsified.

If a Cisco interface is configured with the **no ip directed−broadcast** command, directed broadcasts that would otherwise be "exploded" into link−layer broadcasts at that interface are dropped instead. Note that this means that **no ip directed−broadcast** must be configured on every interface of every router that might be connected to a target subnet; it is not sufficient to configure only firewall routers. The **no ip**

directed−broadcast command is the default in Cisco IOS software version 12.0 and later. In earlier versions, the command should be applied to every LAN interface that isn't known to forward legitimate directed broadcasts.

For a strategy that will block smurf attacks on some firewall routers, depending on the network design, and for more general information on the "smurf" attack, please refer to some information on denial of service attacks .

Path Integrity

Many attacks depend on the ability to influence the paths datagrams take through the network. If they control routing, crackers may be able to spoof the address of another user's machine and have the return traffic sent to them, or they may be able to intercept and read data intended for someone else. Routing could also be disrupted purely for denial of service purposes.

IP Source Routing

The IP protocol supports source routing options that allow the sender of an IP datagram to control the route that datagram will take toward its ultimate destination, and generally the route that any reply will take. These options are rarely used for legitimate purposes in real networks. Some older IP implementations do not process source−routed packets properly, and it may be possible to crash machines running these implementations by sending them datagrams with source routing options.

A Cisco router with **no ip source−route** set will never forward an IP packet which carries a source routing option. You should use this command unless you know that your network needs source routing.

ICMP Redirects

An ICMP redirect message instructs an end node to use a specific router as its path to a particular destination. In a properly functioning IP network, a router will send redirects only to hosts on its own local subnets, no end node will ever send a redirect, and no redirect will ever be traversed more than one network hop. However, an attacker may violate these rules; some attacks are based on this. It's a good idea to filter out incoming ICMP redirects at the input interfaces of any router that lies at a border between administrative domains, and it's not unreasonable for any access list that's applied on the input side of a Cisco router interface to filter out all ICMP redirects. This will cause no operational impact in a correctly configured network.

Note that this filtering prevents only redirect attacks launched by remote attackers. It's still possible for attackers to cause significant trouble using redirects if their host is directly connected to the same segment as a host that's under attack.

Routing Protocol Filtering and Authentication

If you're using a dynamic routing protocol that supports authentication, it's a good idea to enable that authentication. This prevents some malicious attacks on the routing infrastructure, and can also help to prevent damage caused by misconfigured "rogue" devices on the network.

For the same reasons, service providers and other operators of large networks are generally well advised to use route filtering (with the **distribute−list in** command) to prevent their routers from accepting clearly incorrect routing information. Although excessive use of route filtering can destroy the advantages of dynamic routing, judicious use often helps to prevent unpleasant results. For example, if you are using a dynamic routing protocol to communicate with a "stub" customer network, you should not accept any routes from that customer other than routes to the address space you have actually delegated to the customer.

Detailed instruction on configuring routing authentication and route filtering is beyond the scope of this document. Documentation is available on Cisco's Web site and elsewhere. Because of the complexity involved, novices are advised to seek experienced advice before configuring these features on important networks.

Flood Management

Many denial of service attacks rely on floods of useless packets. These floods congest network links, slow down hosts, and can overload routers as well. Careful router configuration can reduce the impact of such floods.

An important part of flood management is being aware of where performance bottlenecks lie. If a flood is overloading a T1 line, then filtering out the flood on the router at the source end of the line will be effective, whereas filtering at the destination end will have little or no effect. If the router itself is the most overloaded network component, then filtering "protections" that place heavy demands on the router are likely to make matters worse. Keep this in mind when you consider implementing the suggestions in this section.

Transit Floods

It is possible to use Cisco's quality of service (QoS) features to protect hosts and links against some kinds of floods. Unfortunately, a general treatment of this sort of flood management is beyond the scope of this document, and the protection depends heavily on the attack. The only simple, generally applicable advice is to use weighted fair queueing (WFQ) wherever CPU resources can support it. WFQ is the default for low−speed serial lines in recent versions of Cisco IOS software. Other features of possible interest include committed access rate (CAR), generalized traffic shaping (GTS), and custom queuing. It's sometimes possible to configure these features when under active attack.

If you do plan to use QoS features to control floods, it's important to understand how those features work, and how common flooding attacks work. For example, WFQ is much more effective against ping floods than against SYN floods, because the usual ping flood will appear to WFQ as a single traffic flow, whereas each packet in a SYN flood will generally appear as a separate flow. A "smurf" reply stream falls somewhere between the two. A great deal of information about Cisco QoS features is available on Cisco's World Wide Web site, and information about common attacks is available at many Web sites maintained by other parties.

Cisco provides two different router features intended specifically to reduce the impact of SYN flooding attacks on hosts. The "TCP Intercept" feature is available in certain software versions for many routers with model numbers of 4000 or greater. The Cisco IOS Firewall Feature Set, which is becoming available on an increasing number of Cisco routers, includes a different SYN flood protection feature. SYN flood protection can be complex, and results may vary depending on flood rate, router speed and memory size, and the hosts in use. If you configure either of these features, you should be sure to read the documentation on Cisco's World Wide Web site, and you should, if possible, test your configuration under an actual flood.

Router Self−protection

Before a router can protect other parts of the network from the effects of floods, the router itself must be protected from overload.

Switching Modes and Cisco Express Forwarding

The CEF switching mode, available in Cisco IOS software versions 11.1CC, 11.1CT, 11.2GS, and 12.0, replaces the traditional Cisco routing cache with a data structure that mirrors the entire system routing table.

Because there is no need to build cache entries when traffic starts arriving for new destinations, CEF behaves more predictably than other modes when presented with large volumes of traffic addressed to many destinations.

Although most flooding denial of service attacks send all of their traffic to one or a few targets and therefore do not tax the traditional cache maintenance algorithm, many popular SYN flooding attacks use randomized source addresses. The host under attack replies to some fraction of the SYN flood packets, creating traffic for a large number of destinations. Routers configured for CEF therefore perform better under SYN floods (directed at hosts, not at the routers themselves) than do routers using the traditional cache. CEF is recommended when available.

Scheduler Configuration

When a Cisco router is fast−switching a large number of packets, it is possible for the router to spend so much time responding to interrupts from the network interfaces that no other work gets done. Some very fast packet floods can cause this condition. The effect can be reduced by using the **scheduler interval** command, which instructs the router to stop handling interrupts and attend to other business at regular intervals. A typical configuration might include the command **scheduler interval 500**, which indicates that process−level tasks are to be handled no less frequently than every 500 milliseconds. This command very rarely has any negative effects, and should be a part of your standard router configuration unless you know of a specific reason to leave it out.

Many newer Cisco platforms use the command **scheduler allocate** instead of **scheduler interval**. The **scheduler allocate** command takes two parameters: a period in microseconds for the system to run with interrupts enabled, and a period in microseconds for the system to run with interrupts masked. If your system doesn't recognize the **scheduler interval 500** command, try **scheduler allocate 3000 1000**. These values were chosen to represent the midpoints of the ranges. The range for the first value is 400 to 60000, and the range for the second value is 100 to 4000. These parameters can be tuned.

Possibly Unnecessary Services

As a general rule, any unnecessary service should be disabled in any router that's reachable from a potentially hostile network. The services listed in this section are sometimes useful, but should be disabled if they aren't actively being used.

TCP and UDP "Small Services"

By default, Cisco devices up through IOS version 11.3 offer the "small services": echo, chargen, and discard. These services, especially their UDP versions, are infrequently used for legitimate purposes, but can be used to launch denial of service and other attacks that would otherwise be prevented by packet filtering.

For example, an attacker might send a DNS packet, falsifying the source address to be a DNS server that would otherwise be unreachable, and falsifying the source port to be the DNS service port (port 53). If such a packet were sent to the Cisco's UDP echo port, the result would be the Cisco sending a DNS packet to the server in question. No outgoing access list checks would be applied to this packet, since it would be considered to be locally generated by the router itself.

Although most abuses of the small services can be avoided or made less dangerous by anti−spoofing access lists, the services should almost always be disabled in any router which is part of a firewall or lies in a security−critical part of the network. Since the services are rarely used, the best policy is usually to disable them on all routers of any description.

The small services are disabled by default in Cisco IOS 12.0 and later software. In earlier software, they may be disabled using the commands **no service tcp−small−servers** and **no service udp−small−servers**.

Finger

Cisco routers provide an implementation of the "finger" service, which is used to find out which users are logged into a network device. Although this information isn't usually tremendously sensitive, it can sometimes be useful to an attacker. The "finger" service may be disabled with the command **no service finger**.

NTP

The Network Time Protocol (NTP) isn't especially dangerous, but any unneeded service may represent a path for penetration. If NTP is actually used, it's important to explicitly configure trusted time source, and to use proper authentication, since corrupting the time base is a good way to subvert certain security protocols. If NTP isn't being used on a particular router interface, it may be disabled with the interface command n**o ntp enable**.

CDP

Cisco Discovery Protocol (CDP) is used for some network management functions, but is dangerous in that it allows any system on a directly−connected segment to learn that the router is a Cisco device, and to determine the model number and the Cisco IOS software version being run. This information may in turn be used to design attacks against the router. CDP information is accessible only to directly connected systems. The CDP protocol may be disabled with the global configuration command **no cdp running**. CDP may be disabled on a particular interface with **no cdp enable**.

Staying Up To Date

Like all software, Cisco IOS software has bugs. Some of these bugs have security implications. In addition, new attacks are always being invented, and behavior that could reasonably have been considered correct when a piece of software was written may have bad effects when deliberately exploited.

When a major new security vulnerability is found in a Cisco product, Cisco generally issues an advisory notice about the vulnerability. For information about the process through which these notices are issued, please refer to Cisco Product Security Incident Response. The notices themselves are available from Cisco Product Security Advisories and Notices.

Almost any unexpected behavior of any piece of software may create a security exposure somewhere, and only bugs with especially direct implications for system security are mentioned in advisories. Your security will be enhanced if you try to keep your software up to date even in the absence of any security advisory.

Some security problems aren't caused by software bugs, and it's important for network administrators to stay aware of trends in attacks. A number of World Wide Web sites, Internet mailing lists, and Usenet newsgroups are concerned with this.

Command List

This section is intended to serve as a reminder of the configuration suggestions in the other sections. Cisco IOS configuration command names are used in the following table as mnemonic aids. Always read the documentation for any command before using it.

Use	To
enable secret	Configure a password for privileged router access.
service password−encryption	Provide a minimum of protection for configured passwords.
no service tcp−small−servers **no service udp−small−servers**	Prevent abuse of the "small services" for denial of service or other attacks.
no service finger	Avoid releasing user information to possible attackers.
no cdp running **no cdp enable**	Avoid releasing information about the router to directly−connected devices.
no ntp enable	Prevent attacks against the NTP service.
no ip directed−broadcast	Prevent attackers from using the router as a "smurf" amplifier.
transport input	Control which protocols can be used by remote users to connect interactively to the router's VTYs or to access its TTY ports.
ip access−class	Control which IP addresses can connect to TTYs or VTYs. Reserve one VTY for access from an administrative workstation.
exec−timeout	Prevent an idle session from tying up a VTY indefinitely.
service tcp−keepalives−in	Detect and delete "dead" interactive sessions, preventing them from tying up VTYs.
logging buffered ***buffer−size***	Save logging information in a local RAM buffer on the router. With newer software, the buffer size may be followed with an urgency threshold.
ip access−group ***list*** **in**	Discard "spoofed" IP packets. Discard incoming ICMP redirects.
ip verify unicast rpf	Discard "spoofed" IP packets in *symmetric routing environments* with CEF only.
no ip source−route	Prevent IP source routing options from being used to spoof traffic.

access−list ***number action criteria*** **log** **access−list** ***number action criteria*** **log−input**	Enable logging of packets that match specific access list entries. Use **log−input** if it's available in your software version.
scheduler−interval **scheduler allocate**	Prevent fast floods from shutting down important processing.
ip route 0.0.0.0 0.0.0.0 null 0 255	Rapidly discard packets with invalid destination addresses.
distribute−list ***list*** **in**	Filter routing information to prevent accepting invalid routes.
snmp−server community ***something−inobvious*** **ro** ***list*** **snmp−server community** ***something−inobvious*** **rw** ***list***	Enable SNMP version 1, configure authentication, and restrict access to certain IP addresses. Use SNMP version 1 only if version 2 is unavailable, and watch for sniffers.
snmp−server party... authentication md5 ***secret ...***	Configure MD5−based SNMP version 2 authentication. Enable read−write access unless you need it. Enable SNMP only if it's needed in your network, and don't configure SNMP only if it's needed in your network.
ip http authentication ***method***	Authenticate HTTP connection requests (if you've enabled HTTP on your router).
ip http access−class ***list***	Further control HTTP access by restricting it to certain host addresses (if you've enabled HTTP on your router).
banner login	Establish a warning banner to be displayed to users who try to log into the router.

Related Information

- **Essential IOS Features Every ISP Should Consider**
- **The Latest in Denial of Service Attacks: "Smurfing"**
- **Cisco Product Security Incident Response**
- **Cisco Security Advisories**
- **RFC 2267**
- **Technical Support − Cisco Systems**

All contents are Copyright © 1992−2003 Cisco Systems, Inc. All rights reserved. Important Notices and Privacy Statement.