



Network System Architects
Engineering Solutions For Business Problems

SECURING CISCO ROUTERS

Steven Kieffer
Network System Architects, Inc.
<http://www.nsai.net>

© Network System Architects, Inc.
November 2002

COPYRIGHT NOTICE

Network System Architects, Inc. (NSAi)

This document and the information contained here is the subject of copyright and intellectual property rights under international convention. All rights reserved. No part of this document may be reproduced or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, or stored in any retrieval system of any nature, without the prior written permission of Network System Architects, Inc. (NSAi).

Network System Architects, Inc. (NSAi)

1550 Larimer St., Suite 222
Denver, CO 80202

TABLE OF CONTENTS

1. INTRODUCTION	5
2. SECURING ACCESS CONTROL TO CONFIGURATION METHODS	6
2.1. Console Port, Auxiliary Port & VTYs.....	6
2.1.1. Securing the Console Port.....	6
2.1.2. Securing the Auxiliary Port	6
2.1.3. Securing the VTY Ports	7
2.2. Securing the HTTP Interface	8
2.3. Securing SNMP	8
2.4. Securing TFTP	9
3. IOS USER PASSWORDS	11
4. IOS USER & COMMAND PRIVILEGE LEVELS.....	12
4.1. Recommended Privilege Level Changes.....	12
5. SECURING PASSWORDS STORED ON THE ROUTER	13
6. AAA (AUTHENTICATION, AUTHORIZATION, ACCOUNTING).....	14
6.1. Choosing an Access Control Server.....	14
6.1.1. TACACS+	14
6.1.2. RADIUS	14
6.1.3. Kerberos	14
6.2. TACACS+ Configuration.....	14
6.2.1. Authentication	15
6.2.2. Authorization.....	15
6.3. RADIUS Configuration.....	16
6.3.1. Authentication	16
6.3.2. Authorization.....	16
7. LOGGING.....	18
7.1. Remote Logging	18
7.1.1. Syslog	18
7.1.2. AAA Accounting.....	19
7.1.3. SNMP.....	20

7.2.	Local Logging	20
7.2.1.	Buffered Logging	20
7.2.2.	Console Logging.....	20
7.2.3.	Terminal Logging.....	21
7.3.	NTP (Network Time Protocol).....	21
8.	ANTI-SPOOFING	22
8.1.	Unicast Reverse Packet Forwarding (uRPF).....	22
8.2.	Anti-spoofing with ACLs.....	22
9.	SECURING THE ROUTING PROTOCOL WITH AUTHENTICATION	24
9.1.	OSPF.....	24
9.2.	EIGRP	24
9.3.	RIPv2.....	25
9.4.	BGP4.....	25
10.	HSRP AUTHENTICATION.....	26
11.	OTHER SERVICES AND PROTOCOLS	27
11.1.	Small Services.....	27
11.2.	Finger	27
11.3.	BootP.....	27
11.4.	Proxy ARP	27
11.5.	Directed Broadcast.....	28
11.6.	Cisco Discovery Protocol (CDP).....	28
11.7.	Source Routing.....	28
11.8.	Classless Routing.....	28
11.9.	Configuration Auto Loading	28
11.10.	ICMP	29
11.10.1.	Deny All ICMP	29
11.10.2.	ICMP Unreachables.....	29
11.10.3.	ICMP Mask Reply.....	29
11.10.4.	ICMP Redirects	29
12.	LOGIN BANNERS	31

1. INTRODUCTION

Lately in security, it seems that the attention has been focused on perimeter and server security. Less attention seems to be given to the security of the actual network infrastructure. Central to the network infrastructure are its routers and routing protocols. This paper focuses on securing your Cisco routers. Cisco IOS provides many options for configuration and this document will guide you in configuring it securely.

2. SECURING ACCESS CONTROL TO CONFIGURATION METHODS

A Cisco router can be accessed and configured in many ways. The main access points are out-of-band through the console port and auxiliary ports, and in-band through TTY (VTY), HTTP, TFTP and SNMP. Each of these access points need to be either configured properly for secure access control or disabled. Also, user authorization for accessing, configuring and running commands should be configured as well so you can control who is logging in and what they can do. This section details each of the configuration methods. It is strongly recommended, though, that VTY access via SSH (which is explained in this section) be the only in-band configuration method available on the router. All others should be disabled for security concerns that will also be described in this section.

2.1. Console Port, Auxiliary Port & VTYS

The console port, auxiliary port and virtual TTYs are called “lines” and provide interactive access to the router. By default, these lines grant user level access. Once in this user mode, an administrator then can escalate his privileges. Securing these ports is a first line of defense.

2.1.1. Securing the Console Port

The console port is used to access the router directly. It is an out-of-band management method that is usually accessed via a serial cable and a terminal emulation program. This port has the unique feature of allowing an administrator to recover or reset the router’s password. While this feature is powerful and useful, it also makes its physical security of utmost importance. A modem should never be attached to this port because it could be possible for an attacker to reset the systems passwords and gain full control over the router if he is somehow able to force the system to reload.

You should assign a password to the console port; issue the following commands from the privileged, “enable” mode:

```
NSAi-Cisco(config)#line console 0
NSAi-Cisco(config-line)#login
NSAi-Cisco(config-line)#password securepassword
```

Issue the following commands from the privileged, “enable” mode to disable logins to the console port:

```
NSAi-Cisco(config)#line console 0
NSAi-Cisco(config-line)#login
NSAi-Cisco(config-line)#no password
```

Please note that disabling logins to the console port does not protect against an attacker reloading the system and performing a password recovery. Controlling physical access to the router is the only way of protecting against this.

2.1.2. Securing the Auxiliary Port

The auxiliary port is also used for out-of-band management. A modem or terminal server is usually attached to this port for remote access. Connecting a modem with a public phone line directly to this port makes it available for attack from anyone who has access to the public phone system and can figure out its phone number. It is highly recommended that a terminal server be used instead to control access to this port. Either way, this port should be assigned a login password.

Issue the following commands from the privileged, “enable” mode to assign a password to the auxiliary port.

```
NSAi-Cisco(config)#line aux 0
NSAi-Cisco(config-line)#login
NSAi-Cisco(config-line)#password securepassword
```

Issue the following commands from the privileged, “enable” mode to disable logins to the auxiliary port.:

```
NSAi-Cisco(config)#line aux 0
NSAi-Cisco(config-line)#login
NSAi-Cisco(config-line)#no password
```

2.1.3. Securing the VTY Ports

The VTY ports are virtual TTY ports and are used for in-band management of the device. By default, they can be reached from anywhere on the network. Because of this, special attention must be given to the access control placed on these ports. It is strongly recommended that all other in-band management methods be disabled and that VTY via SSH be the only in-band management access permitted. Explanations and examples follow.

By default, IOS provides five TTY ports. Issue the following commands from the privileged, “enable” mode to assign a password to all five auxiliary ports simultaneously:

```
NSAi-Cisco(config)#line vty 0 4
NSAi-Cisco(config-line)#login
NSAi-Cisco(config-line)#password securepassword
```

By default, IOS allows for many protocols to connect to these TTY ports. Depending on the version of IOS, telnet, rlogin and SSH might all be usable for connection. It is recommended that clear text authentication protocols not be used, as it might be possible for an attacker to “sniff” the passwords traversing the network. If your version of IOS supports IPsec, it is recommended that the VTY ports be configured to only accept SSH connections. With this configured, connections would be made using SSH clients such as SecureCRT and Putty.

There are several steps required to enable this feature:

If it is not already, assign the router’s hostname and domain names.. This is done with the ‘hostname’ and ‘ip domain-name’ commands

Encryption keys must be generated. The ‘crypto key generate rsa’ command will ask you for a modulus size. The recommended a minimum size is 1024. A larger size will increase security but might hinder performance.

SSH must be enabled (time out and authentication retry values must be set)

Each line must be configured to use SSH.

The following commands accomplish this and assume that the router already has a hostname and domain assigned:

```
NSAi-Cisco(config)#crypto key generate rsa
NSAi-Cisco(config)#ip ssh time-out 60
NSAi-Cisco(config)#ip ssh authentication-retries 2
NSAi-Cisco(config)#line vty 0 4
NSAi-Cisco(config-line)#transport input ssh
```

By default, the VTY ports are accessible from any IP address that can reach the router. This makes it easy for a person to conduct his attack from anywhere on your network. It is recommended that access to these VTY ports be limited to specific hosts or networks using access lists. For example, the following commands will create an access list to only permit hosts 172.27.1.5 and 172.27.2.7 access to the VTY ports. The log argument found at the end of the deny ACL statement allows you to see denied connection attempts. See the Logging section for further details.

```
NSAi-Cisco(config)#access-list 10 permit 172.27.1.5
NSAi-Cisco(config)#access-list 10 permit 172.27.2.7
NSAi-Cisco(config)#access-list 10 deny any any log
NSAi-Cisco(config)#line vty 0 4
NSAi-Cisco(config-line)#access-class 10 in
```

2.2. Securing the HTTP Interface

Cisco has added the ability to remotely manage and monitor routers via a standard web browser. This access method is similar to an interactive session with the router and is of high security concern. This access method works over HTTP and inherits the insecurities of the HTTP protocol. All content, including usernames and passwords, are transmitted across the network in clear text. In addition, by default, the HTTP web access uses the enable password to log in and grants the user privileged level rights.

IOS does not enable the HTTP service by default. The clear text authentication and the power of the interface make this a large security risk. Because of this risk, it is recommended that HTTP router access not be employed.

Despite the security risks, some administrators may choose to use the HTTP management interface. If so, a few measures can be taken to improve its security.

The first recommendation is that it be disabled. The following commands will ensure that HTTP access is disabled.

```
NSAi-Cisco(config)#no ip http server
```

The following commands can be use to enable the HTTP interface:

```
NSAi-Cisco(config)#ip http server
```

By default, the HTTP interface is accessible from any IP address that can reach the router. This makes it easy for a person to conduct his attack from anywhere on your network. Access to the HTTP interface should be limited to specific IP addresses. This is accomplished using the following commands. The log argument found at the end of the deny ACL statement allows you to see denied connection attempts. See the Logging section for more details.

```
NSAi-Cisco(config)#access-list 20 permit 172.27.1.5
NSAi-Cisco(config)#access-list 20 deny any log
NSAi-Cisco(config)#ip http access-class 20
```

As mentioned earlier, the HTTP interface uses the enable password to grant access by default and grants privileged level access. This is extremely risky because if an HTTP login is captured, the enable password is revealed. HTTP authentication should be configured to use other methods such as local accounts, TACACS+ or AAA. Using these methods, privilege levels can be limited via user accounts. Account privilege level restrictions will be explained later in this document. Use the following commands to configure the HTTP interface:

```
NSAi-Cisco(config)#ip http authentication (local | tacacs | aaa)
```

2.3. Securing SNMP

Cisco devices running IOS have very capable SNMP (Simple Network Management Protocol) agents. SNMP allows for remote monitoring and management of a router. SNMP uses a form of password called community strings for authentication. With the Read-Only (Public) community string, statistics and configuration information can be gained. The Read/write (Private) community string adds the ability to reconfigure the router. Unfortunately, SNMP is a very insecure protocol. Like the HTTP management method, the SNMP v1 and v2 transport the community strings in clear text. There is a new SNMP v3

standard in the works that takes care of the security flaws. However, at the time of this writing, SNMP v1 and v2 are the only choices.

As is the recommendation for HTTP access, it is recommended that SNMP access be disabled. Use the following command to disable SNMP access:

```
NSAi-Cisco(config)#no snmp-server
```

If SNMP must be employed, a few things can be done to increase security. First, treat community strings like the passwords they are and choose strings that are not easily guessable. All too commonly, administrators use community strings of “public” for read access and “private” for read/write access. Attackers often try these common strings in their guessing attempts so it is very important that they not be used. Second, if SNMP will be used solely for statistics gathering and monitoring, Read-Write access should be disabled.

The following commands set Read-Only and Read-Write SNMP passwords on a Cisco device:

```
NSAi-Cisco(config)#snmp-server community password RO
NSAi-Cisco(config)#snmp-server community differentpassword RW
```

The following commands sets Read-Only password and disables the Read-Write SNMP functionality on a Cisco device:

```
NSAi-Cisco(config)#snmp-server community password RO
NSAi-Cisco(config)#no snmp-server community differentpassword RW
```

For increased SNMP security, apply an access list to limit SNMP access to the router. The following commands create an access list that limits access only from 10.128.128.1 and applies it to Read-Write SNMP access. The log argument found at the end of the deny ACL statement allows you to see denied connection attempts. See the Logging section for further details.

```
NSAi-Cisco(config)#access-list 3 permit 10.128.128.1
NSAi-Cisco(config)#access-list 3 deny any any log
NSAi-Cisco(config)#snmp-server community differentpassword RW 3
```

SNMP community strings are usually sent in plain text across the network and are therefore vulnerable to sniffing.

SNMP uses UDP as a transport protocol, which makes it much easier for an attacker to impersonate a valid source IP address. Theoretically, an attacker could gain Read-Only SNMP access by creating packets with a source IP address allowed by the access list and then sniffing the reply packets. Alternatively, an attacker could gain Read-Write access by creating packets with a source IP address allowed by the access list and simply ignore the reply packets.

Support for the many flavors of SNMP security varies greatly between different releases of IOS. Check the Cisco web site for documentation on the specific release of IOS you are working with.

2.4. Securing TFTP

Cisco routers use TFTP (Trivial File Transfer Protocol) to transfer IOS code and configuration files in and out of the device. The TFTP protocol does not provide for any authentication measures. This is not a big risk when transfers are initiated from the router by administrators to upgrade software or import/export configuration files.

The risks are more serious when a router is configured to download its configuration from a remote TFTP server during system boot. If an attacker is able to compromise or spoof the TFTP server and force the router to reboot, he can load whatever configuration he chooses.

Newer versions of IOS allow the router to be configured as a TFTP server. It is a serious security risk to have this enabled. When enabled, an attacker can connect to the router and download the configuration file. It should be noted that this feature is disabled by default but you can explicitly disable it by issuing the following command:

```
NSAi-Cisco(config)#no tftp-server
```

3. IOS USER PASSWORDS

The IOS default method of authentication relies on passwords only. This provides no accountability, as there are no usernames involved, everyone logs in with the same password. If there is to be more than one person administering the device it is recommended that either an access control server or local username access control be employed. This will provide separate accounts for individuals logging into the device. Using an access control server is the preferred option and will be detailed in the AAA section of this document. This section describes local username access.

First, create username/password combinations, using the following example:

```
NSAi-Cisco(config)#username skieffer password stevespass  
NSAi-Cisco(config)#username wspencer password willspass
```

Second, local authentication is tied to each line. Following is an example of enabling local authentication for the VTYS. This can be applied similarly to all lines (console, aux and VTY)

```
NSAi-Cisco(config)#line vty 0 4  
NSAi-Cisco(config-line)#login local
```

4. IOS USER & COMMAND PRIVILEGE LEVELS

Cisco routers have two levels of authorization by default, level 1 and level 15. Level 1 authorization is commonly called user mode and only allows users to view information about the router and use a few simple utilities such as ping. Level 1, or user mode, does not allow changes to be made to the configuration of the router. Level 15 authorization is commonly called privileged mode and allows full, unrestricted rights to reconfigure and view everything about the router. These two levels do not provide for much granularity. IOS provides the ability to custom define 16 different levels, 0-15. It also provides the ability to assign these custom privilege levels to commands and users. This allows for fine-grained authorization of certain users to run certain commands. Two steps are required to enable this functionality.

First, use the following commands to assign privilege levels to user accounts:

```
NSAi-Cisco(config)#username skieffer privilege 15
NSAi-Cisco(config)#username wspencer privilege 5
```

Second, assign privilege levels to commands. Following are examples of the commands required to assign privilege 15 to telnet and privilege 4 to ping:

```
NSAi-Cisco(config)#privilege exec level 15 telnet
NSAi-Cisco(config)#privilege exec level 4 ping
```

In the example above, the user skieffer would be able to execute the telnet program and the user wspencer would not.

4.1. Recommended Privilege Level Changes

By default, many commands are available at privilege level 1. These include telnet, rlogin, connect, show access-lists, show ip access-lists and show logging. All these commands are useful to an attacker and are available, by default, to anyone who gains user-level access. It is recommended that the privilege level of these commands be changed to level 15. The commands for doing so are listed below:

```
NSAi-Cisco(config)#privilege exec level 15 telnet
NSAi-Cisco(config)#privilege exec level 15 rlogin
NSAi-Cisco(config)#privilege exec level 15 connect
NSAi-Cisco(config)#privilege exec level 15 show access-lists
NSAi-Cisco(config)#privilege exec level 15 show ip access-lists
NSAi-Cisco(config)#privilege exec level 15 show logging
NSAi-Cisco(config)#privilege exec level 1 show ip
```

The last line listed above, 'privilege exec level 1 show ip', is required to allow the rest of the default level 1 commands to function properly.

5. SECURING PASSWORDS STORED ON THE ROUTER

IOS relies primarily on passwords for authentication. IOS passwords can be stored either in the device's configuration or on a separate authentication server such as TACACS+ or RADIUS. The most secure way to handle passwords is to maintain them off of the router on a TACACS+ or RADIUS server. This method is explained in detail later in the Access Control Servers section of this document.

It is often necessary to maintain passwords in the router's configuration. By default, IOS stores these passwords in clear-text. Commands must be issued to instruct IOS to encrypt them. There are two levels of password encryption that IOS supports, Cisco's Password 7 hashing and MD5 hashing.

IOS can be instructed to go through its configuration file and encrypt clear-text passwords by issuing the service password-encryption command.

Enable `Service password-encryption` using the following command:

```
NSAi-Cisco(config)#service password-encryption
```

If `service password-encryption` is configured on the IOS device, most passwords are encrypted using the weak password 7 encryption algorithm. This algorithm is weak, and a knowledgeable attacker can determine the plain text password from the encrypted password. It is intended to prevent the casual viewer of a configuration from easily learning the system's passwords. Password 7 crackers do exist that are capable of instantly reversing the hash, revealing the password.

Because of this weakness Cisco offers much stronger encryption for the privileged, enable password. This method employs the non-reversible MD5 hashing algorithm. Enable MD5 hashing for the enable password by using the following command:

```
NSAi-Cisco(config)#enable secret password
```

Even though the MD5 is non-reversible, it is still susceptible to a brute force or dictionary attack. As such, it is still important to keep the configuration file away from untrusted people.

6. AAA (AUTHENTICATION, AUTHORIZATION, ACCOUNTING)

AAA is a Cisco access control and accounting feature that is built in to IOS. The AAA architecture is extremely robust and can authenticate, authorize and account for many forms of network usage. These forms of usage include things such as controlling dial-up users' access to certain network resources and logging the duration of the connection and bytes transferred for billing purposes.

As this paper is intended to help you secure your Cisco router, only aspects of AAA that pertain to securing the router will be covered. This includes the remote storage of user accounts and passwords, user authorization to run commands and the logging of events and changes. The logging (accounting) portion of AAA will be described in the Logging section of this document.

AAA works with a security database for the storage of user accounts, passwords, credentials and logs. This database can reside either locally or on a remote access control server. The local security database arrangement offers very limited AAA functionality and does not offer much in the way of increased router security.

The recommended setup is to employ an access control server. This method provides the benefits of a central, secure user account, credential and password store as well as allowing for comprehensive logging.

6.1. Choosing an Access Control Server

Cisco currently supports three remote security database standards for use with AAA: TACACS+, RADIUS and Kerberos. Cisco offers an access control server called Cisco Secure ACS. It runs on both Solaris and Windows platforms and supports both the TACACS+ and RADIUS protocols. There are also free versions TACACS+ and RADIUS. A quick search on the Web will find you many links to them.

6.1.1. TACACS+

TACACS+ is the recommended choice for a Cisco only environment. Its drawback is that it is Cisco proprietary and cannot interact with non-Cisco gear. TACACS+ has several design features that make it theoretically more secure than RADIUS. TACACS+ communicates via TCP as opposed to UDP with RADIUS. TACACS+ also encrypts the entire payload of the packets it communicates over while RADIUS only encrypts the passwords contained in the packets.

6.1.2. RADIUS

RADIUS is the recommended choice for environments where interoperability with non-Cisco gear is required. It is an industry standard protocol defined by Internet standards and supported by all major network equipment vendors.

6.1.3. Kerberos

Although Kerberos is an option, it is not recommended for use with AAA. Kerberos provides only for authentication and does not provide for authorization or accounting. Because of these limitations, Kerberos will not be discussed further.

6.2. TACACS+ Configuration

Configuration of the TACACS+ server and user accounts is out of the scope of this document. Reference the server's documentation for more information. Following are instructions for IOS configuration to enable AAA support for authentication, authorization and accounting with TACACS+.

6.2.1. Authentication

TACACS+ configuration for user mode is very simple. These commands configure TACACS+ authentication from the TACACS+ server at 172.27.1.14 for VTY connections 0 through 4, the console and auxiliary ports with a failover to local user authentication, should the TACACS+ server be unavailable.

```
NSAi-Cisco(config)#aaa new-model
NSAi-Cisco(config)#tacacs-server host 172.27.1.14
NSAi-Cisco(config)#tacacs-server key SharedKeyWithServer
NSAi-Cisco(config)#aaa authentication login default group tacacs+ local
NSAi-Cisco(config)#line vty 0 4
NSAi-Cisco(config-line)#login authentication default
NSAi-Cisco(config-line)#exit
NSAi-Cisco(config)#line con 0
NSAi-Cisco(config-line)#login authentication default
NSAi-Cisco(config-line)#exit
NSAi-Cisco(config)#line aux 0
NSAi-Cisco(config-line)#login authentication default
```

Although the use of the HTTP interface is still not recommended, it may also employ TACACS+ authentication. The configuration command follows and assumes you have already performed the “aaa” and “tacacs-server” commands listed previously:

```
NSAi-Cisco(config)#ip http authentication aaa
```

TACACS+ configuration for privileged mode can also be simple. This command configures TACACS+ authentication for the `enable` command and also assumes you have already performed the “aaa” and “tacacs-server” commands listed previously:

```
NSAi-Cisco(config)#aaa authentication enable default group tacacs+ enable
```

6.2.2. Authorization

There are two types of authorization to set for the purpose of securing the router, exec authorization and command authorization.

Exec authorization grants or denies authenticated users access to the IOS command line shell (EXEC prompt). Configure exec authorization by using the following command:

```
NSAi-Cisco(config)#aaa authorization exec default group tacacs+ if-authenticated
```

The ‘if-authenticated’ is a common optional argument and allows EXEC prompt access to authenticated users should the TACACS+ server be unavailable. To protect against a possible attack where the TACACS+ server is intentionally made unreachable, substitute ‘if-authenticated’ with ‘none’.

Command authorization instructs the router to check with the TACACS+ server for user authorization to run commands. It is configured per command privilege level. The following commands instructs the router to check for authorization for all privilege level 1 and level 15 commands:

```
NSAi-Cisco(config)#aaa authorization commands 1 default group tacacs+ if-
authenticated
NSAi-Cisco(config)#aaa authorization commands 15 default group tacacs+ if-
authenticated
```

Again, the ‘if-authenticated’ is a common optional argument and allows EXEC prompt access to authenticated users should the TACACS+ server be unavailable. To protect against a possible attack where the TACACS+ server is intentionally made unreachable, substitute ‘if-authenticated’ with ‘none’.

6.3. RADIUS Configuration

Configuration of the RADIUS server and user accounts is out of the scope of this document. Reference the server's documentation for more information. Following are instructions for IOS configuration to enable AAA support for authentication, authorization and accounting with RADIUS.

6.3.1. Authentication

The configuration of RADIUS in IOS uses very similar commands to the configuration of TACACS+. The only difference is that the word "radius" is used in place of tacacs and tacacs+. The following commands configure RADIUS authentication from the RADIUS server at 172.27.1.15 for VTY connections 0 through 4, the console and auxiliary ports with a failover to local user authentication, should the TACACS+ server be unavailable.

```
NSAi-Cisco(config)#aaa new-model
NSAi-Cisco(config)#radius-server host 172.27.1.15
NSAi-Cisco(config)#radius-server key SharedKeyWithServer
NSAi-Cisco(config)#aaa authentication login default group radius local
NSAi-Cisco(config)#line vty 0 4
NSAi-Cisco(config-line)#login authentication default
NSAi-Cisco(config-line)#exit
NSAi-Cisco(config)#line con 0
NSAi-Cisco(config-line)#login authentication default
NSAi-Cisco(config-line)#exit
NSAi-Cisco(config)#line aux 0
NSAi-Cisco(config-line)#login authentication default
```

Although the use of the HTTP interface is still not recommended, it may also employ RADIUS authentication. The configuration command follows and assumes you have already performed the "aaa" and "tacacs-server" commands listed previously:

```
NSAi-Cisco(config)#ip http authentication aaa
```

RADIUS configuration for privileged mode can also be simple. This command configures RADIUS authentication for the 'enable' command and also assumes you have already performed the "aaa" and "radius-server" commands listed previously:

```
NSAi-Cisco(config)#aaa authentication enable default group radius enable
```

6.3.2. Authorization

There are two types of authorization to set for the purpose of securing the router, exec authorization and command authorization.

Exec authorization grants or denies authenticated users access to the IOS command line shell (EXEC prompt). Use the following command to configure authorization:

```
NSAi-Cisco(config)#aaa authorization exec default group radius if-authenticated
```

The 'if-authenticated' is a common optional argument and allows EXEC prompt access to authenticated users should the TACACS+ server be unavailable. To protect against a possible attack where the RADIUS server is intentionally made unreachable, substitute 'if-authenticated' with 'none'.

Command authorization instructs the router to check with the RADIUS server for user authorization to run commands. It is configured per command privilege level. The following commands instructs the router to check for authorization for all privilege level 1 and level 15 commands:

```
NSAi-Cisco(config)#aaa authorization commands 1 default group radius if-  
authenticated  
NSAi-Cisco(config)#aaa authorization commands 15 default group radius if-  
authenticated
```

Again, the 'if-authenticated' is a common optional argument and allows EXEC prompt access to authenticated users should the RADIUS server be unavailable. To protect against a possible attack where the TACACS+ server is intentionally made unreachable, substitute 'if-authenticated' with 'none'.

7. LOGGING

Logging is important for detecting an attack that is in the works and for figuring out what happened after an attack. If you monitor your logs in a real time fashion, you may be able to detect an attack that is mounting; for example, you may notice a port scan in progress or critical settings being changed. Command logging is very important in determining what has been or what is being done to the router. In addition, by piecing together logs from routers distributed throughout your network, you can observe things from a great vantage point.

Cisco router categorizes its log messages into severity levels from 0 to 7 with the lower number designating higher severity. The following table lists these severity levels:

Level	Level Name	Syslog Definition	Description
0	emergencies	LOG_EMERG	System unusable
1	alerts	LOG_ALERT	Immediate action needed
2	critical	LOG_CRIT	Critical conditions
3	errors	LOG_ERR	Error conditions
4	warnings	LOG_WARNING	Warning conditions
5	notifications	LOG_NOTICE	Normal but significant
6	informational	LOG_INFO	Informational messages
7	debugging	LOG_DEBUG	Debugging messages

There are many ways a Cisco router can log information. These methods can be categorized under remote logging and local logging.

7.1. Remote Logging

Remote logging is the preferred method of logging for several reasons. Logs stored remotely are preferred because log entries can be more detailed and held on a secure remote server. Also, with all routers reporting to a central log store, log analysis and trending is more easily facilitated. Automated software can be used to analyze these centrally stored logs and report on its findings. Three Cisco remote logging methods that can be configured on a router are syslog, AAA accounting and SNMP traps. The recommended methods of logging are syslog combined with AAA accounting. With this combination, you have the most detailed and complete logging solution.

7.1.1. Syslog

With syslog logging, a syslog server is set up somewhere on the network with the router(s) configured to log to it. Configuring a syslog server is out of the scope of this document. However, the following commands configure the router side settings:

The following commands cause the IOS device to send logging messages to two syslog servers for redundancy:

```
NSAi-Cisco(config)#logging 10.128.128.1
NSAi-Cisco(config)#logging 10.128.128.2
```

Modify the messages sent to a syslog server normally use the syslog local7 facility by using the `logging facility` command:

```
NSAi-Cisco(config)#logging facility local6
```

Log messages should be timestamped to allow you to correlate logs from different devices on your network and to help you make a case against an attacker in court if that becomes necessary. The following commands cause debugging and logging messages to be timestamps with the local date and time, including the time zone, with accuracy to the millisecond.

```
NSAi-Cisco(config)#service timestamps log datetime msec localtime show-timezone
```

Sequence numbers should also be stamped with each log entry. This helps an administrator to figure out if the log files have been tampered with. If your version of IOS supports this feature, enable it with the following command:

```
NSAi-Cisco(config)#service sequence-numbers
```

It is convenient to force syslog messages to be stamped with one IP address, regardless of which interface of the router the originated from. This usually makes it easier to search your log files.

```
NSAi-Cisco(config)#logging source-interface loopback0
```

It is also important to configure your syslog server to only accept log entries from the IP addresses of configured routers. Although this can be defeated through spoofing, it adds an additional layer of protection.

7.1.2. AAA Accounting

AAA accounting provides additional detail beyond what syslog logging can log. AAA accounting can log items such as who logged into the router, from where and for how long and what commands were performed by whom, when. When AAA accounting is combined with syslog logging, you have the most complete logging solution. The following AAA accounting configuration requires that a TACACS+ or RADIUS server be set up as described in the AAA section of this document.

The command syntax for enabling AAA accounting follows:

```
aaa accounting {system|network|exec|connection|commands level}  
{default|listname}  
{start-stop|wait-start|stop-only|none} group {tacacs+|radius}
```

- system - logs all system events
- network - logs all network service requests
- exec - logs all router EXEC commands
- connection - logs all outbound connections (ex. Telnet)
- commands *level* - logs all commands of a given level
- default – indicates that the accounting methods specified serve as the default list
- listname – is an assigned name to a list of accounting methods previously defined (not used in our examples)
- start-stop – the router sends an accounting notice to the security server when a process starts and when a process stops
- wait-start – the router sends an accounting notice to the security server but does not start the service until it receives confirmation from the security server that it received the notice.
- stop-only – the router sends an accounting notice to the security server at when the service stops
- none – disables accounting (seems pointless!)

- tacacs+ - designates the TACACS+ server as the destination for accounting data
- radius - designates the RADIUS server as the destination for accounting data

The following commands configure AAA accounting to recommended settings. Note that the words “tacacs” and “tacacs+” in the following configuration may be replaced with the word “radius” if a radius server is being employed:

```
NSAi-Cisco(config)#aaa new-model
NSAi-Cisco(config)#tacacs-server host 172.27.1.14
NSAi-Cisco(config)#tacacs-server key SharedKeyWithServer
NSAi-Cisco(config)#aaa accounting exec default start-stop group tacacs+
NSAi-Cisco(config)#aaa accounting system default stop-only group tacacs+
NSAi-Cisco(config)#aaa accounting connection default start-stop group tacacs+
NSAi-Cisco(config)#aaa accounting network default start-stop group tacacs+
NSAi-Cisco(config)#aaa accounting command 1 default start-stop group tacacs+
NSAi-Cisco(config)#aaa accounting command 15 default start-stop group tacacs+
```

7.1.3. SNMP

Although remote logging with both syslog and AAA accounting is preferred, you may already have an SNMP management station that you wish to log to through traps. Send traps containing information on configuration changes and down interfaces to a management station for logging using the following commands:

```
NSAi-Cisco(config)#snmp-server host 172.27.1.15
NSAi-Cisco(config)#snmp-server enable traps
```

7.2. Local Logging

As mentioned in the beginning of this section, remote logging is the preferred logging method. Local logging is only recommended in situations where remote logging is not practical. Even then, of the three local logging options, buffered, console and terminal, only buffered logging is recommended.

7.2.1. Buffered Logging

Buffered logging stores log data, although temporarily, in the router’s RAM. There are some drawbacks to using this method. The log buffer size is limited so when full, old messages are deleted as new entries are made. In addition, since these entries are stored in RAM, all data would be lost should the router be cycled. Buffered logging is only recommended if remote logging is not possible.

The following commands turn logging on, sets the log buffer size in RAM to 32K and sets the router to log messages of informational severity or higher. Depending on the amount of RAM in the router, this size may be increased. You must figure out a balance between having a large enough buffer without cutting into the router’s performance through RAM deprivation.

```
NSAi-Cisco(config)#logging on
NSAi-Cisco(config)#logging buffered 32000
NSAi-Cisco(config)#logging buffered informational
```

View buffered logs by issuing the ‘show log’ commands.

7.2.2. Console Logging

Console logging logs information to the console port. This is almost useless for security, as someone would have to be hired full time to stare at a connected terminal, read and make sense of the scrolling data.

7.2.3. Terminal Logging

Terminal logging is just as useless for security. In this case, the person hired to stare at the terminal has the convenience of doing so from a VTY session.

7.3. NTP (Network Time Protocol)

No matter what your method of logging, it is important to have an accurate timestamp of when things happened. This becomes even more important when you start correlating log events from different systems. These systems need to have a similar account of time so you can begin to accurately piece things together. This is where NTP comes in. With NTP, all routers on the network reference a central time source for synchronization.

It is possible that an attacker could compromise your NTP system by spoofing your network timeserver and injecting inaccurate time into your systems. This is why it is recommended that your NTP clients and servers employ authentication.

Although complex NTP configurations in IOS are possible with routers serving as NTP servers, this document will describe a simple configuration with a central time server. These instructions assume you have an existing NTP server on your network at IP 172.27.1.20 capable of exchanging md5 authentication keys.

```
NSAi-Cisco(config)#ntp authenticate
NSAi-Cisco(config)#ntp authentication-key 1 md5 KeySharedWithNTPServer
NSAi-Cisco(config)#ntp trusted-key 1
NSAi-Cisco(config)#ntp server 172.27.1.20 key 1
```

8. ANTI-SPOOFING

Attackers often employ a technique known as spoofing to defeat access control security measures. An example of this would be an attacker on the outside of your network sending altered packets to your router with a source address belonging to your internal network. Without anti-spoofing measures in place, this attack could bypass your access control, allowing access to the router and into your internal network. It is recommended that anti-spoofing measures be put in place on routers that connect to external networks to prevent such an attack. Cisco routers have two ways to protect against spoofing. One is called Unicast Reverse Packet Forwarding (uRPF) and the other is accomplished through ACL configuration. URPF is the preferred method of anti-spoofing because it results in less of a performance hit than using ACLs, it dynamically configures itself to changing network topologies and it is simple to configure.

There are additional anti-spoofing measures that can be taken to further protect your network (route filtering, etc.). As this paper is focused on securing the router, such a discussion would be out of scope

8.1. Unicast Reverse Packet Forwarding (uRPF)

URPF is an intelligent IOS feature that uses its knowledge of your network to make decisions on whether to deny or allow packets with a particular source address into a particular interface. URPF relies on another Cisco feature called Cisco Express Forwarding (CEF) to function. Configuration consists of globally enabling CEF, then applying the 'ip verify unicast reverse-path' command to each interface you wish to perform anti-spoofing on.

```
NSAi-Cisco(config)#ip cef
NSAi-Cisco(config)#interface serial 0/1
NSAi-Cisco(config-if)#ip verify unicast reverse-path
```

That's it!

One drawback to using uRPF is that it does not understand asymmetrical routing and can cause problems. For this reason it is recommended that uRPF be used only on routers and interfaces that connect to external networks or internally where asymmetrical routing does not occur.

8.2. Anti-spoofing with ACLs

If it is determined that you cannot use uRPF, then an alternative is to antispoof using ACLs. This method is a little more complex to set up, does not have the flexibility to automatically adapt to your changing network and can take more of a performance hit on your router.

To antispoof using ACLs, you first create an access list denying all internal networks and allowing all other traffic. You then apply that list inbound on an external interface. It is also good practice to also include all RFC 1918 private networks, loopback addresses, broadcast networks and multicast networks as these are commonly used in attacks. Adding the log argument to your deny access list entries is recommended and to enable the logging of address spoofing attempts. Example commands follow and assume your internal network is 182.12.34.0/24 and your external interface is Serial 0/1.

```
NSAi-Cisco(config)#access-list 10 deny 182.12.34.0 0.0.0.255 log
NSAi-Cisco(config)#access-list 10 deny 10.0.0.0 0.255.255.255 log
NSAi-Cisco(config)#access-list 10 deny 127.0.0.0 0.255.255.255 log
NSAi-Cisco(config)#access-list 10 deny 172.16.0.0 0.15.255.255 log
NSAi-Cisco(config)#access-list 10 deny 192.168.0.0 0.0.255.255 log
NSAi-Cisco(config)#access-list 10 deny 224.0.0.0 15.255.255.255 log
NSAi-Cisco(config)#access-list 10 deny 240.0.0.0 7.255.255.255 log
NSAi-Cisco(config)#access-list 10 deny 255.255.255.255 0.0.0.0 log
NSAi-Cisco(config)#access-list 10 permit any
NSAi-Cisco(config)#interface serial 0/1
NSAi-Cisco(config-if)#ip access-group 10 in
```

9. SECURING THE ROUTING PROTOCOL WITH AUTHENTICATION

As you most likely know, routing protocols are how routers communicate with each other and determine the routes that network traffic will take. An attacker can inject bad routes into a routing protocol causing it to send network traffic along a path of his choosing. To protect against this it is important that all your routing protocols be secured by configuring them to use authentication. Commands to configure Cisco routers to use MD5 authentication with OSPF, EIGRP, BGP4 and RIPv2 will be detailed in this section. All communicating routing devices must be configured with the same passwords for authentication to work.

Please note that this document does not, in any way, attempt to be a complete manual for the listed routing protocols. Each protocol has had entire books written on them for you to read. This section provides simple configuration steps that can be taken to increase security.

9.1. OSPF

Set passwords on each interface that will participate in OSPF communications by configuring the MD5 authentication in OSPF with the following commands:

```
NSAi-Cisco(config)#interface ethernet 0
NSAi-Cisco(config-if)#ip ospf message-digest-key 1 md5 YourPasswordHere
NSAi-Cisco(config-if)#exit
NSAi-Cisco(config)#interface ethernet 1
NSAi-Cisco(config-if)#ip ospf message-digest-key 1 md5 YourPasswordHere
```

Second, issue the following commands to tell the router the areas you wish to use authentication. The following commands perform this assuming an OSPF autonomous system 20:

```
NSAi-Cisco(config)#router ospf 20
NSAi-Cisco(config-router)#area 0 authentication message-digest
```

9.2. EIGRP

EIGRP uses the concept of a keychain to store passwords in on the router. First, configure the keychain with your password using the following commands:

```
NSAi-Cisco(config)#key chain chain1
NSAi-Cisco(config-keychain)#key 1
NSAi-Cisco(config-keychain-ke)#key-string YourPasswordHere
```

Second, configure each interface that will participate in EIGRP communications to use MD5 authentication with the password you placed in the keychain called "chain1". The following commands perform this assuming an EIGRP autonomous system 20:

```
NSAi-Cisco(config)#interface ethernet 0
NSAi-Cisco(config-if)#ip authentication mode eigrp 20 md5
NSAi-Cisco(config-if)#ip authentication key-chain eigrp 20 chain1
NSAi-Cisco(config-if)#exit
NSAi-Cisco(config)#interface ethernet 1
NSAi-Cisco(config-if)#ip authentication mode eigrp 20 md5
NSAi-Cisco(config-if)#ip authentication key-chain eigrp 20 chain1
```

9.3. RIPv2

RIPv2 also uses the concept of a keychain to store passwords in on the router. First, configure the keychain with your password using the following commands:

```
NSAi-Cisco(config)#key chain chain1
NSAi-Cisco(config-keychain)#key 1
NSAi-Cisco(config-keychain-ke)#key-string YourPasswordHere
```

Second, configure each interface that will participate in EIGRP communications to use MD5 authentication with the password you placed in the keychain called “chain1”.

```
NSAi-Cisco(config)#interface ethernet 0
NSAi-Cisco(config-if)#ip rip authentication key-chain chain1
NSAi-Cisco(config-if)#ip rip authentication mode md5
NSAi-Cisco(config-if)#exit
NSAi-Cisco(config)#interface ethernet 1
NSAi-Cisco(config-if)#ip rip authentication key-chain chain1
NSAi-Cisco(config-if)#ip rip authentication mode md5
```

9.4. BGP4

BGP4 configuration for using MD5 authentication is accomplished by simply adding the command ‘password’ to the end of the neighbor statement and defining the password.

```
NSAi-Cisco(config)#router bgp 300
NSAi-Cisco(config-router)#neighbor 182.12.34.15 password YourPasswordHere
```

10. HSRP AUTHENTICATION

If your router participates in an HSRP configuration, it should be configured to use authentication. Although authentication passwords are exchanged in clear text, it can increase security. It might be possible for an attacker to configure an unauthorized router in such a way that it could take over routing functions for a network. Using HSRP authentication would make this more difficult. Configure the participating interfaces by issuing the following commands:

```
NSAi-Cisco(config)#interface ethernet 0
NSAi-Cisco(config-if)#standby authentication password
```

11. OTHER SERVICES AND PROTOCOLS

Every service running on the router can give a hacker another avenue of attack. IOS enables some services and protocols by default. Often, these services are not used and run unnecessarily. If these services are not needed, it is recommended that they be turned off.

11.1. Small Services

IOS provides a number of small TCP/IP services similar to those provided by the `inetd` process on most Unix systems. The Cisco defined small TCP services are `echo`, `chargen`, `discard`, and `daytime`. The Cisco defined small UDP services are `echo`, `chargen`, and `discard`. In IOS versions prior to 11.3, small services are enabled by default should be disabled using the following commands:

```
NSAi-Cisco(config)#no service udp-small-servers
NSAi-Cisco(config)#no service tcp-small-servers
```

11.2. Finger

The finger service is an administrative utility that allows a user to remotely see who is currently logged into the router. Attackers use this feature to gather valid login names to use in his hacking attempts. Finger is enabled by default and should be disabled using the following command:

```
NSAi-Cisco(config)#no service finger
```

11.3. BootP

BootP is a service that allows a router to boot itself off of an IOS image located on another router. This can also allow an attacker to download a copy of your IOS image. BootP is enabled by default and should be disabled by issuing the following command:

```
NSAi-Cisco(config)#no ip bootp server
```

11.4. Proxy ARP

Proxy ARP is a feature that allows a node that is not configured with a default gateway to send packets outside of its network. With this feature enabled, a router replies to a node's ARP request for the MAC address of a remote node with its own MAC address. It enables transparent access between multiple network segments. Because of this, it can break the LAN security perimeter. Proxy ARP is enabled by default. It is recommended that be disabled if you do not need it. Issue the following commands on a per interface basis to enable proxy ARP:

```
NSAi-Cisco(config)#interface serial 0
NSAi-Cisco(config-if)#no ip proxy-arp
```

11.5. Directed Broadcast

A directed broadcast allows a node on one network to send a broadcast to a different network. This technique is used in denial of service attacks. Because of this, it is important for routers to reject directed broadcast packets. Directed broadcasts are enabled by default in IOS 11.3 and earlier. It should be disabled and is done so on a per interface basis using the following commands:

```
NSAi-Cisco(config)#interface serial 0
NSAi-Cisco(config-if)#no ip directed-broadcast
```

11.6. Cisco Discovery Protocol (CDP)

CDP is a feature that is included on all Cisco equipment that allows devices to find out information about other Cisco devices to which they are connected. An attacker might be able to use this information to aid in his efforts. It is recommended that it be disabled if it is not needed. CDP is enabled by default and can be disabled globally by issuing the following command:

```
NSAi-Cisco(config)#no cdp run
```

CDP can also be disabled on a per interface basis by issuing the following commands:

```
NSAi-Cisco(config)#interface serial 0
NSAi-Cisco(config-if)#no cdp enable
```

11.7. Source Routing

Source routing is a feature that allows a packet to determine the routing path it will take through the network. This feature can be used to bypass your network's normal routing paths. An attacker can use this to route his traffic around your access control points and intrusion detection systems. Source Routing is enabled by default. If you do not need source routing, it is recommended that it be disabled using the following command:

```
NSAi-Cisco(config)#no ip source-route
```

11.8. Classless Routing

With classless routing enabled, a router will forward packets even if it does not have a solid route defined for its destination. This feature is enabled by default, can aid in certain attacks, and should be disabled if it is not needed by your routing scheme. Do not disable this if you are employing classless routing. The following command disables this feature:

```
NSAi-Cisco(config)#no ip classless
```

11.9. Configuration Auto Loading

Cisco routers have the ability to automatically their configuration file from a remote TFTP server. Remote loading of configuration is not considered secure as an attacker now has the opportunity to spoof the TFTP server and load a configuration of his choosing. This feature is disabled by default and can be explicitly disabled using the following commands:

```
NSAi-Cisco(config)#no boot network
NSAi-Cisco(config)#no service config
```

11.10. ICMP

ICMP (Internet Control Message Protocol) is a very helpful protocol that provides utility to aid in network troubleshooting. Attackers also find ICMP to be a very helpful protocol to aid in their attack. ICMP provides various services, some of which are dangerous and should be disabled.

11.10.1. Deny All ICMP

It is always best to deny all ICMP access coming into your network from an untrusted network. It is best to block all ICMP however; it is necessary to still allow ICMP Type 3 Code 4 packets through. These packets deal with IP's MTU discovery functions and are necessary for the normal operation of IP through the router. The following commands create an ACL that blocks all ICMP packets except Type 3 Code 4 and applies it to interface Serial 0:

```
NSAi-Cisco(config)#access-list 105 permit icmp any any 3 4
NSAi-Cisco(config)#access-list 105 deny icmp any any
NSAi-Cisco(config)#access-list 105 permit ip any any
NSAi-Cisco(config)#interface serial 0
NSAi-Cisco(config-if)#ip access-group 105 in
```

11.10.2. ICMP Unreachables

When a host sends a packet to a host or network that a router does not know about, an ICMP unreachable message is returned. This is a courteous feature that lets the sending host know that his attempt failed. This same feature can also speed an attacker's mapping of your network because his scanner does not have to wait for the attempt to time out. Your router quickly responds with an ICMP unreachable message. This feature is enabled by default and should be disabled at on external interfaces facing untrusted networks. Disable ICMP unreachables on a per interface basis by issuing the following commands:

```
NSAi-Cisco(config)#interface serial 0
NSAi-Cisco(config-if)#no ip unreachables
```

11.10.3. ICMP Mask Reply

This feature allows the router to tell a requesting host what the correct subnet mask is for a given network. This aids an attacker in mapping your network. ICMP Mask reply is not commonly used and is disabled by default. It should not be used and can be explicitly disabled on a per interface basis by issuing the following commands:

```
NSAi-Cisco(config)#interface serial 0
NSAi-Cisco(config-if)#no ip mask-reply
```

11.10.4. ICMP Redirects

An IP redirect, sent to an end node instructs that node to use a particular router to get to a particular destination. The original intended use of this feature was for a router to send redirects only to hosts on its directly connected networks.

It is possible for an attacker to take advantage of this feature and send an ICMP redirect message instructing nodes to send all traffic through a router of his choosing. ICMP redirects are enabled by default and should be disabled. Disabling ICMP redirects should not cause any problems if your network employs a routing protocol. It is recommended that it be disabled on all interfaces of all routers or at a minimum, on any external interfaces facing an untrusted network. Configure the router to block both the sending and receiving of ICMP redirects.

To disable the router from sending ICMP redirects on the interface Serial 0:

```
NSAi-Cisco(config)#interface serial 0
NSAi-Cisco(config-if)#no ip redirects
```

To disable the router from receiving ICMP redirects you must employ an ACL to the interfaces. The following commands configures such an ACL to interface Serial 0:

```
NSAi-Cisco(config)#access-list 100 deny icmp any any redirect
NSAi-Cisco(config)#access-list permit ip any any
NSAi-Cisco(config)#interface serial 0
NSAi-Cisco(config-if)#ip access-group 101 in
```

12. LOGIN BANNERS

Login banners are important for router security, not from the technical standpoint but from a legal one. If a properly worded banner is not presented upon login, it may not be possible to pursue legal action against an attacker. The banner must express that:

- Only authorized users are to use the system.
- The system is to be used for official work only.
- All use and access may be monitored, recorded, and presented to the appropriate officials.
- The use of this system implies that you consent to the above conditions
- A legal expert should examine the actual wording you choose.

Banners should be configured to display when a user attempts to login. Use the following commands to configure the banner:

```
NSAi-Cisco(config)#banner login $  
Banner content  
Banner content  
Banner content  
$
```