8 steps to protect your Cisco router

Daniel B. Cid
daniel@underlinux.com.br

Network security is a completely changing area; new devices like IDS (Intrusion Detection systems), IPS (Intrusion Prevention systems), and Honeypots are modifying the way people think about security. Companies are spending thousand of dollars on new security devices, but forgetting the basic, the first line of defense: the border router.

Although a lot of people may think that routers don't need to be protect, they are completely wrong. A lot of secure problems appear all time against this kind of device and most of them are vulnerable.
Some information about some common security problems found on Cisco Routers, can be read on the text "Exploiting Cisco Routers", available at:
http://www.securityfocus.com/infocus/1734

In this article I will give you 8 steps, easy to follow, to minimize your Cisco router exposure by turning off some unused services, applying some access control and applying some security options available on that.

1- **Control Access to your router;**
2- **Restrict telnet access to it;**
3- **Block Spoof/Malicious packets;**
4- **Restrict SNMP;**
5- **Encrypt all passwords;**
6- **Disable all unused services;**
7- **Add some security options;**
8- **Log everything;**

1- **Control Access to your router**

The first thing to do is apply some rules to restrict all external access to some ports of the router. You can block all ports, but it is not always necessary. These commands bellow will protect your router against some reconnaissance attacks and, obviously, will restrict access to these ports:

```
access-list 110 deny tcp any host $yourRouterIP eq 7
access-list 110 deny tcp any host $yourRouterIP eq 9
access-list 110 deny tcp any host $yourRouterIP eq 13
```

```
access-list 110 deny tcp any host $yourRouterIP eq 19
access-list 110 deny tcp any host $yourRouterIP eq 23
access-list 110 deny tcp any host $yourRouterIP eq 79
int x0/0
access-group in 110
```

Where $yourRouterIP is your router IP and x0/0 is your external interface. We Will always use this convention in this article.

2- **Restrict telnet access to it**

Telnet is not a very safe protocol to use, but if you really need to use it (you should always use ssh) you might want to restrict all access to it (remember that all your traffic will be unencrypted). The best way to accomplish that is using a standard access-list and the *access-class* command.

```
access-list 50 permit 192.168.1.1
access-list 50 deny any log
line vty 0 4
access-class 50 in
exec-timeout 5 0
```

Where 192.168.1.1 is the IP address allowed to telnet the router

3- **Block Spoof/Malicious packets**

You must never allow loopback/reserved IP address from the Internet reach your external interface and you can reject broadcast and multicast addresses too.

```
access-list 111 deny ip 127.0.0.0 0.255.255.255 any
access-list 111 deny ip 192.168.0.0 0.0.0.255 any
access-list 111 deny ip 172.16.0.0 0.0.255.255 any
access-list 111 deny ip 10.0.0.0 0.255.255.255 any
access-list 111 deny ip host 0.0.0.0 any
access-list 111 deny ip 224.0.0.0 31.255.255.255 any
access-list 111 deny icmp any any redirect
int x0/0
access-group in 111
```

4- **Restrict SNMP**

SNMP must always be restrict, unless you want some malicious person getting a lot of information from your network ☺

```
access-list 112 deny udp any any eq snmp
access-list 112 permit ip any any
interface x0/0
access-group 112 in
```

And if you are not going to use SNMP at all, disable it:

```
no snmp-server
```

5- **Encrypt all passwords**

A very important thing to do is protect all your passwords using the powerful algorithm as possible.
The password from *exec* mode, that grants privileged access to the IOS system,
Can be set using a MD5 hash, which is the strongest option available on the
Cisco IOS.

```
enable secret $yourpassword
```

All other passwords, you can encrypt using the Vigenere cipher that is not
Very strong, but can help. To do that, you can use the *service password-encryption*
Command that encrypts all passwords present in you system.

```
service password-encryption
```

6- **Disable all unused services**

6.1 - **Disable Echo, Chargen and discard**

```
no service tcp-small-servers
no service udp-small-servers
```

6.2 - **Disable finger**

```
no service finger
```

6.3 - **Disable the httpd interface**

```
no ip http server
```

6.4 - **Disable ntp (if you are not using it)**

```
ntp disable
```

7- **Add some security options**

7.1 - **Disable source routing**

```
no ip source-route
```

7.2 - **Disable Proxy Arp**

```
no ip proxy-arp
```

7.3 - **Disable ICMP redirects**

```
interface s0/0 (your external interface)
no ip redirects
```

7.4 - **Disable Multicast route Caching**

```
interface s0/0 (your external interface)
no ip mroute-cache
```

6.5 - **Disable CDP**

```
no cdp run
```

6.6 - **Disable direct broadcast (protect against Smurf attacks)**

```
no ip directed-broadcast
```

8- **Log everything**

To finish, you must log everything on an outside Log Server. You must everything from all your systems and always analyze the logs.

```
logging trap debugging
logging 192.168.1.10
```

where 192.168.1.10 is the ip of your log server (configured as a Syslog server)

Conclusion

With these simple steps you can add a lot of security to your router, protecting it against a lot of possible attacks, increasing your network security.

Only as an example, you can see the *nmap* result before and after applying these options:

Before:

```
bash-2.05b# nmap -O 192.168.1.1

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Interesting ports on  (192.168.1.1):
Port       State       Service
7/tcp      open        echo
9/tcp      open        discard
13/tcp      open        daytime
19/tcp      open        chargen
23/tcp      open        telnet
79/tcp      open        finger
80/tcp      open        http
Remote OS guesses: AS5200, Cisco 2501/5260/5300 terminal server IOS 11.3.6(T1),
Cisco IOS 11.3 - 12.0(11)
```

After:

```
bash-2.05b# nmap -P0 -O 192.168.1.1

Starting nmap V. 3.00 ( www.insecure.org/nmap/ )
Warning:  OS detection will be MUCH less reliable because we did not find at least 1
open and 1 closed TCP port
All 1601 scanned ports on  (192.168.1.1) are: filtered
Too many fingerprints match this host for me to give an accurate OS guess

Nmap run completed -- 1 IP address (1 host up) scanned in 403 seconds
```