Interested in learning more about security?

SANS Institute InfoSec Reading Room

Securing The Network With Cisco Router

Proper configuration of routers is important as this will help to resist attacks and ensures the integrity and confidentiality of network traffic. This paper expands on the work of Mark Degner [Ref.1] on securing Cisco routers. In it [Ref.1], Deger discussed about securing Cisco routers from malicious attack through limiting access, securing the remote administration of routers using secure shell and the shutting down of unneeded services provided by the routers. He also went on to cover SYN/smurf attacks protection, p...

Securing The Network With Cisco Router
Author: Bang Shuh Tan
Date: May 18, 2002

Introduction

In a computer network, routers are responsible for controlling much of the flow of data. Hence, proper configuration of routers is important as this will help to resist attacks and ensures the integrity and confidentiality of network traffic.

This paper expands on the work of Mark Degner [Ref.1] on securing Cisco routers. In it [Ref.1], Deger discussed about securing Cisco routers from malicious attack through limiting access, securing the remote administration of routers using secure shell and the shutting down of unneeded services provided by the routers. He also went on to cover SYN/smurf attacks protection, performing ingress/egress filtering and logging. This paper expands upon that by discussing additional steps and security features available on a Cisco router for enhancing the security of anetwork. First, we will cover the securing of routing updates through neighbor router authentication [Ref.2] and route filtering. Next, we will discuss the topic of using IPSec to secure remote administration of Cisco routers. Following that, we will have an overview of reflexive access list and content-based application control. Then, we will touch on combating code red with network-based application recognition. We will end with a short discussion on performing integrity checking on routers. For each of the above discussions, a sample IOS configuration will be shown.

Neighbor Router Authentication

The role of routers in a network is to direct packets to the appropriate destination network. In order to do that, routers have a routing table that contains all routes to all known network. Routers learn about these routes in 2 ways: statically or dynamically. Administrators add static routes into the routing table while dynamic routes are learned from other routers. While static routes are more secure, it is not feasible as it does not scale. Thus, dynamic routing updates are commonly used in large enterprise networks.

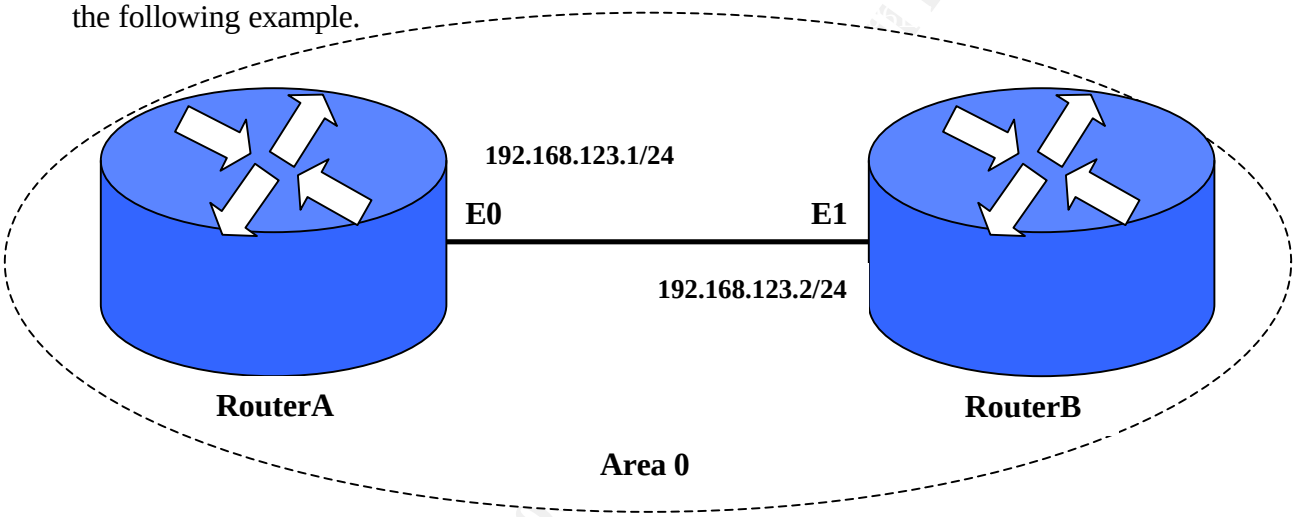
However, using dynamics routing updates come with its security concern. An attacker could send fraudulent routing updates to the routers and in doing so cause a denial of service attack or redirect network traffic for some malicious purpose. To protect the integrity of the routing domain, routers must have some means of identifying that the routing update came from a trusted source. This is achieved through neighbor router authentication [Ref.2]. Please note that neighbor router authentication is not supported in RIP version 1.

In neighbor router authentication, the router authenticates the source of each routing update packet that it receives using a secret key. All routers in the same domain share the same secret key. The sending router sends the routing updates along with the secret key. The receiving router then compares the secret key it receives with its own secret key. If they match, the receiving router will trust the update and add it to its routing table.

There are 2 types of neighbor authentication: plain text authentication and Message Digest Algorithm Version 5 (MD5) authentication. In plain text authentication, the shared secret key is sent along with the routing update in clear text. This method does not provide much security as an attacker could easily obtain the secret key using a packet sniffer. For this reason, plain text authentication should not be used.

In MD5 authentication, the router uses the MD5 algorithm to produce a hash of the secret key. The hash is then send instead of the secret key along with the routing update. This way, even if the attacker manages to sniff the routing update, he/she will be unable to obtain the secret key.

The Cisco IOS® configuration for MD5 authentication for OSPF [Ref.3] is illustrated in the following example.

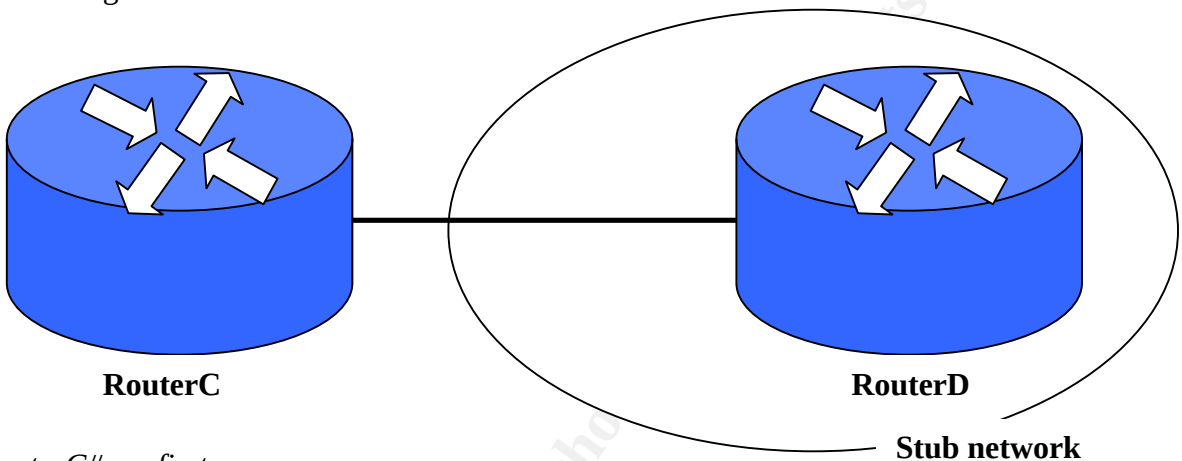


RouterA# config t
RouterA(config)# router ospf 0
RouterA(config-router)# network 192.168.123.0 0.0.0.255 area 0
RouterA(config-router)# area 0 authentication message-digest
RouterA(config-router)# exit
RouterA(config)# int e0
RouterA(config-if)# ip ospf message-digest-key 1 md5 shared-secret-key
RouterA(config-if)# end
RouterA#

RouterB# config t
RouterB(config)# router ospf 0
RouterB(config-router)# network 192.168.123.0 0.0.0.255 area 0
RouterB(config-router)# area 0 authentication message-digest
RouterB(config-router)# exit
RouterB(config)# int e1
RouterB(config-if)# ip ospf message-digest-key 1 md5 shared-secret-key
RouterB(config-if)# end

Route Filtering

Even though enabling neighbor router authentication ensures that the routing updates originated from a trusted source. We could secure the network further using route filtering [Ref.4]. This is in accordance with the defense in depth principle. In route filtering, distribute lists are used to filter/suppress advertisements of routing updates. In our example network below, RIP is used to communicate with a stub network, RouterC should not accept any routes from RouterD other than routes that belong to the address space assigned to the stub network.



RouterC# config t
RouterC(config)# access-list 1 permit 192.168.123.0 0.0.0.255
RouterC(config)# router rip
RouterC(config-router)# distribute-list 1 in

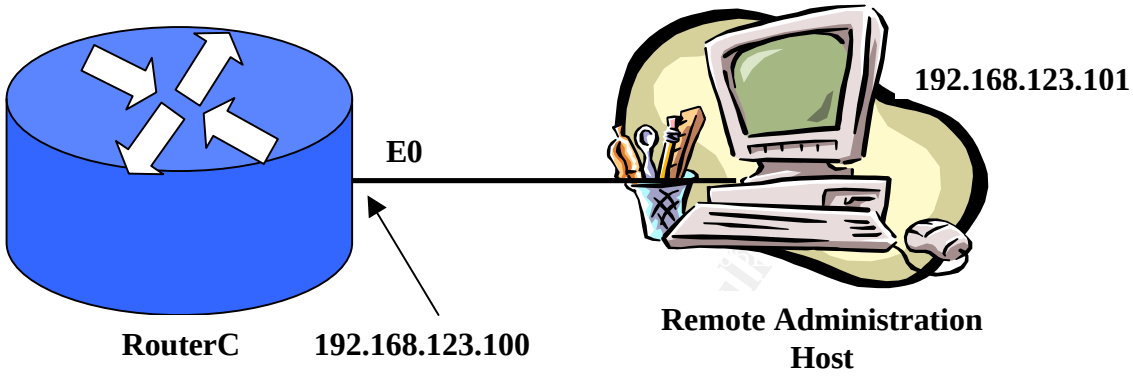
Securing remote administration using IPSec

The danger of network sniffers, couple with the fact that telnet transfers password in clear text give good reason that telnet should not be used to perform remote administration of routers. In [Ref.1], Deger discussed how secure shell could be used instead to encrypt the communication. However, Cisco IOS® only supports secure shell version 1, which is susceptible to multiple vulnerabilities [Ref.5]. These vulnerabilities appear in all Cisco IOS® 12.0 and later releases that include support for SSH. If a network administrator is using secure shell for remote administration, he/she should ensure that the router is running IOS containing the fixes. Also secure shell version 1 may be subjected to man-in-the-middle attack [Ref.6]. Another issue with secure shell is that the inherent design of secure shell means it is open to attacks [Ref.7] at TCP layer.

IPSec is a framework of open standards developed by the Internet Engineering Task Force. It provides confidentiality, integrity and authentication for the transfer of information over a network. IPSec is available in Cisco IOS® Release 11.3(3)T and later. Unlike secure shell, IPSec offers encryption at network layer. A discussion of IPSec is not within the scope of this paper. For more information on IPSec, please refer to Cisco's white paper [Ref.8]. IPSec is generally used in establishing virtual private networks. However, IPSec can also be used for the remote administration of Cisco routers. Another

advantage of IPSec is that it provides an added layer of security for other remote administrations tools like SNMP.

The following example shows the configurations needed for using IPSec for remote administration:



In the above example, interface E0 is assigned an ip address of 192.168.123.100 and the remote administration has an ip address of 192.168.123.101.

We begin by limiting the hosts that can be used to perform remote administration of the router. This is achieved with the help of access list.

RouterC# config t
RouterC(config)# access-list 99 permit 192.168.123.101
RouterC(config)# line vty 0 4
RouterC(config-line)# access-class 99 in
RouterC(config-line)# exit

Next, we setup the ISAKMP policy [Ref .9].

RouterC(config)# crypto isakmp policy 1
RouterC(config-isakmp)# authentication pre-share
RouterC(config-isakmp)# hash sha
RouterC(config-isakmp)# encryption 3des
RouterC(config-isakmp)# group 2
RouterC(config-isakmp)# exit
RouterC(config)#

The above configurations setup an ISAKMP policy that uses pre-shared keys with hashing algorithm, SHA and 3DES encryption. DES should not be used because it uses a weak 64 bits block cipher which given today's computer processing power, a brute force attack can be successfully carry out in a matter of days. The *group 2* statement specifies the type of Diffie-Hellman group. Group 2 is more secure as it uses 1024 bits but it is computational more expensive. The priority number of 1 identifies the IKE policy.

Then we specify the pre-shared key and the ip address of the remote administration host. Make sure to use a hard to guess key.

RouterC(config)# crypto isakmp key hard-to-guess-password address 192.168.123.101

After which we specify the transform-set used for protecting the network traffic [Ref.10].

RouterC(config)# crypto ipsec transform-set routerc esp-3des esp-sha-hmac
RouterC(cfg-crypto-trans)# mode tunnel
RouterC(cfg-crypto-trans)# exit
RouterC(config)#

In the above, we specify a transform-set routerc to operate in tunnel mode as it is more secure. Also ESP header is used as it provides confidentiality, integrity and authentication. The encryption used is 3DES and the authentication method, SHA.

Following that, we specify an extended access control list that identifies the traffic type that will be protected using IPSec.

RouterC(config)# access-list 101 permit ip host 192.168.123.100 host 192.168.123.101 log

Next, we establish a crypto map [Ref.10]. An IPSec crypto map *routercmap* is created using extended access-list 101 to determine if the traffic needs to be protected. The peer statement specifies the remote IPSec peer to which IPSec protected traffic should be forwarded. The transform-set routerc is applied to the crypto map.

RouterC(config)# crypto map routercmap 10 ipsec-isakmp
RouterC(config-crypto-map)# set peer 192.168.123.101
RouterC(config-crypto-map)# set transform-set routerc
RouterC(config-crypto-map)# match address 101
RouterC(config-crypto-map)# exit
RouterC(config)#

Finally, the crypto map is applied on to the interface.

RouterC(config)# int e0
RouterC(config)# crypto map routercmap

Once the router has been configured, the next step is to configure the host for IPSec. Configuration of the host is not within the scope of this paper. For information on configuring IPSec for Windows 2000, please refer to [Ref.11,12].

One thing to note when using IPSec is that it uses IP protocol 50 for ESP and 51 for AH. Also protocol 17, UDP port 500 is used to pass IKE traffic. Ensure these ports are permitted appropriately by the access lists applied to the router.

Reflexive access list

With static standard or extended access lists, session filtering can be performed using the *established* keyword with the *permit* command. This is shown in the example below.

Router(config)# access-list 155 permit tcp any any established

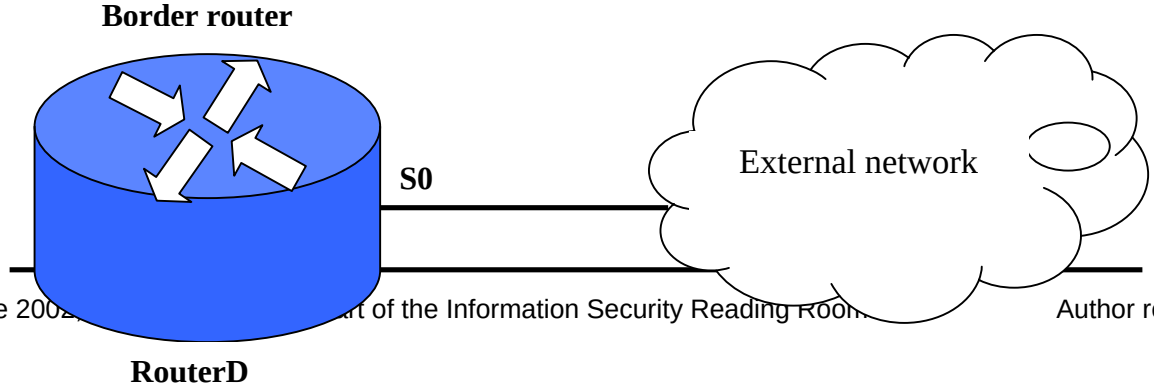
The *established* keyword only filters TCP traffic by checking whether the ACK or RST bits are set. However, this method is not secure and a white paper was written regarding circumventing Cisco access lists which rely on only permitting established TCP sessions by establishing communications between a client and server without using the SYN bit [Ref. 13].

Cisco reflexive access lists available from IOS revisions 11.3 onwards offer a higher level of security. An explanation of reflexive access list:

"Reflexive access lists allow the packet filter to "remember" what it has seen, and then allow only the corresponding response packets back in through the filtering mechanism. To be counted as a response, the incoming packet must be from the host and port to which the outbound packet was sent, and must be directed to the host and port that sent the outbound packet. The router essentially modifies the filtering rules on the fly to accommodate these returning packets." [Ref.14]

Reflexive access lists not only check the ACK or RST bits but both the address and port of the source and destination. Another advantage of reflexive access lists is that it supports other upper layer protocols UDP, ICMP etc. Reflexive access lists should be configured on border routers and can only be defined using named extended access list. Two keywords are used when defining a reflexive access lists: *reflect* and *evaluate*. The *reflect* keyword is used to update a dynamic access list with the mirror image of the packet matching the ACL entry. Returning traffic is later checked against this dynamic ACL using the *evaluate* keyword.

Reflexive access lists can be applied on either internal or external interface depending on the topology [Ref.15]. In the topology below, the reflexive access list is applied to the external interface S0.



First we define an *outbound* IP extended named access list.
RouterD(config)# ip access-list extended outgoing
RouterD(config-ext-nacl)# permit tcp any any reflect trafficlist

Next, we define an *inbound* IP extended named access list.

RouterD(config)# ip access-list extended incoming
RouterD(config-ext-nacl)# evaluate trafficlist

Finally, we apply the access-lists to the external interface.

RouterD(config)# int s0
RouterD(config-if)# ip access-group outgoing out
RouterD(config-if)# ip access-group incoming in

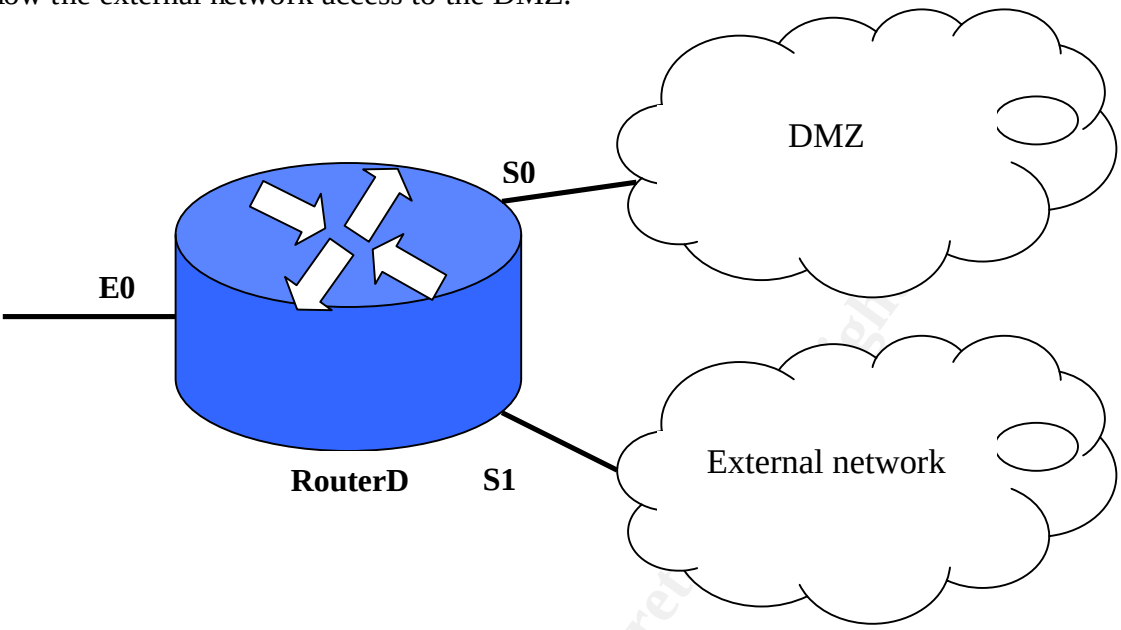
In the above configuration, if a host at 192.168.123.155 sends a tcp packet from source port 1698 to a web server at 172.16.100.10. The following dynamic entry will be added to *incoming* access list:

Permit tcp host 172.16.100.10 eq 80 host 192.168.123.155 eq 1698

There dynamic entries will be removed from the reflexive access list under 2 circumstances. The first is when the TCP session ends via a FIN or RST bit. The second is when there is a lack of traffic and a timeout occurs. The timeout can be adjusted using:

*RouterD(config)# ip reflexive-list timeout **seconds***

In the next topology, the reflexive access list is applied on the internal interface E0 to allow the external network access to the DMZ.



First we define an *outbound* IP extended named access list.

RouterD(config)# ip access-list extended outgoing
RouterD(config-ext-nacl)# evaluate trafficlist

Next, we define an *inbound* IP extended named access list.

RouterD(config)# ip access-list extended incoming
RouterD(config-ext-nacl)# permit tcp any any reflect trafficlist

Finally, we apply the access-lists to the internal interface.

RouterD(config)# int e0
RouterD(config-if)# ip access-group outgoing out
RouterD(config-if)# ip access-group incoming in

For the internal interface, the configurations are similar to the previous example except the incoming and outgoing access list are swapped.

Content-based access control

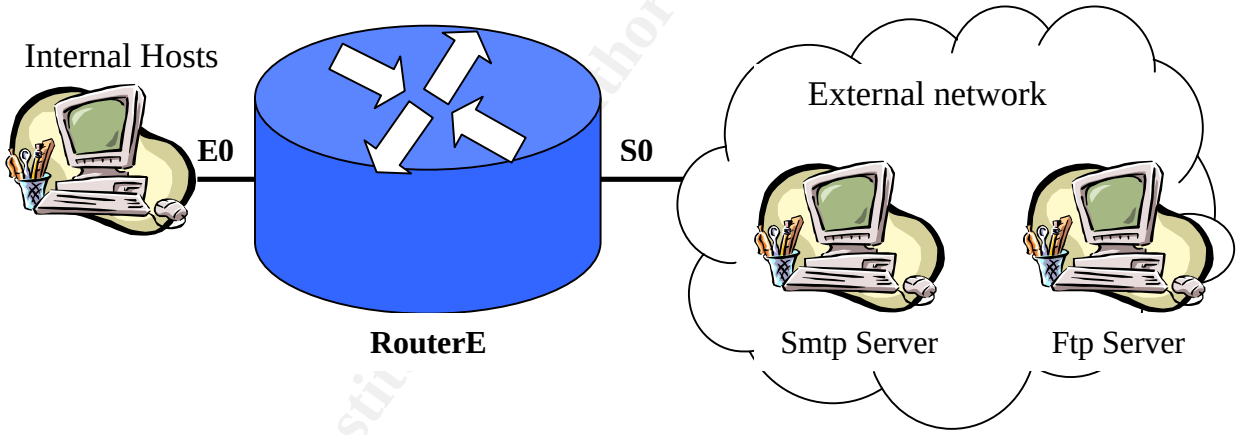
An improvement over reflexive access list is to use Content-based access control (CBAC) of the Cisco IOS® security feature. CBAC is a cheap and effective way of turning an existing Cisco router into a stateful firewall. CBAC is available from Cisco IOS® version 11.2(11)P onwards.

CBAC examines not only network layer and transport layer information, but also examines the application-layer protocol information (such as FTP information) to learn

about the state of TCP and UDP connections. CBAC works by maintaining connection state information for individual connections. The state information is used to make intelligent decisions about whether packets should be permitted or denied, and dynamically modifying the access lists at the firewall interfaces, thereby creating temporary openings in the firewall [Ref.16]. The examining of the IP traffic information is done using the *ip inpect name* command.

CBAC can also be configured to block java applets from unknown or untrusted sites to protect attacks against malicious commands and viruses. Another advantage of CBAC is that it can be used to combat denial-of-service attack. This is done by setting thresholds on the total number of existing half-open sessions and the rate of session establishment attempts. When using CBAC , reflexive access list should not be used.

The following example shows the configurations of CBAC to filter the inbound Ftp and Smtp traffic on the outside interface of a router:



First, we define the inbound access list and apply it to the external interface s0. ICMP is not inspected by CBAC and in order to control the type of ICMP traffic at the interface, a limited set of ICMP traffic can be permitted by using static access list entries.

RouterE(config)# ip acess-list extended 101
RouterE(config-ext-nacl)# permit icmp any any echo-reply
RouterE(config-ext-nacl)# permit icmp any any unreachable
RouterE(config-ext-nacl)# permit icmp any any ttl-exceeded
RouterE(config-ext-nacl)# deny ip any any log
RouterE(config-ext-nacl)# exit
RouterE(config)# int s0
RouterE(config-if)# ip access-group 101 in

RouterE(config-if)# exit

Next, we create the CBAC inspection ruleset. The ruleset specify what IP traffic will be inspected by CBAC on the interface. The *audit-trail* option controls whether use of that protocol causes a log message to be generated.

RouterE(config)# ip inspect name outgoing ftp audit-trail on
RouterE(config)# ip inspect name outgoing smtp audit-trail on

Lastly, we apply the ruleset to S0.

RouterE(config)# int s0
RouterE(config-if)# ip inspect outgoing out
RouterE(config-ifl)# exit

Based on the state information obtained through inspection, CBAC creates a temporary access list entry which is inserted at the beginning of the external interface's inbound extended access list 101. This temporary access list entry is designed to permit inbound packets that are part of the same connection as the outbound packet that was inspected.

Combating "Code Red" worm using Network-based Application Recognition

The "Code Red" worm is a self-replicating malicious code that exploits a known vulnerability in Microsoft IIS servers Version 4 and 5. The worm would send a HTTP GET request to a host that would overflow a buffer in Microsoft's Index Server, a part of IIS. This buffer overflow allowed the worm to execute arbitrary code. Code Red would install a copy of itself into memory on the infected computer, and attempt to infect additional hosts. For more information on Code Red, please refer to Ref. 17.

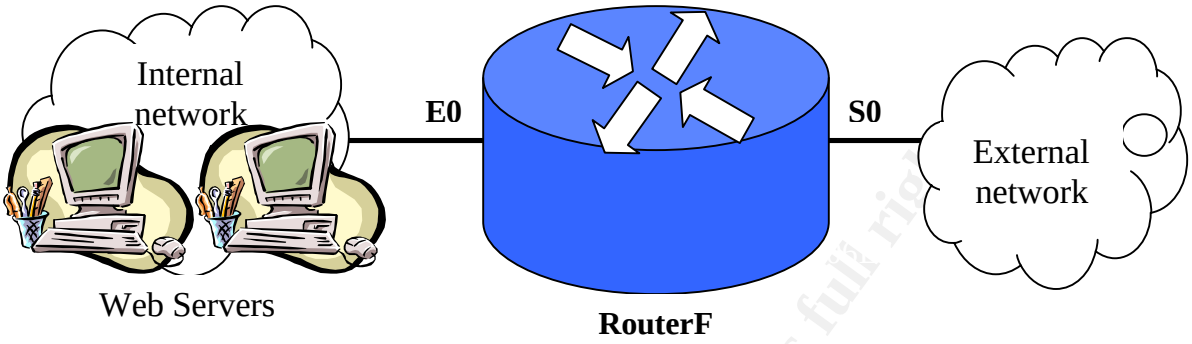
A patch is available from Microsoft, which protects vulnerable system and removes the worm from infected system. However, applying the patch to the servers only prevents the worm from infecting the servers, it does not stop the HTTP GET requests from hitting the servers. There is still the possibility that the server got overwhelmed with a flood of infection attempts.

In accordance with the defense in depth principle, the router can be used in conjunction with the Microsoft's patch by blocking the HTTP GET request at the network ingress point. To prevent infected clients from spreading the code red worm, egress filtering can also be performed at the router.

Network-based Application Recognition (NBAR) together with access list can be used to block the "Code Red" worm at the network ingress point [Ref.18]. NBAR is a classification engine in Cisco IOS® software that can recognize a wide variety of applications, including Web-based applications and client/server applications that dynamically assign Transmission Control Protocol (TCP) or UDP port numbers. Once the

application is recognized, appropriate quality-of-service (QoS) policies can be applied to the traffic classes.

The following example shows the sample configuration for blocking "Code Red" at the network ingress point:



We begin by classifying inbound Code Red traffic with the class-based marking feature in IOS. Since Code Red traffic generate a GET request looking for a file with an .ida extension, we can use it to perform matching:

RouterF(config)#class-map match-any http-hacks
*RouterF(config-cmap)#match protocol http url "*default.ida*"*
*RouterF(config-cmap)#match protocol http url "*cmd.exe*"*
*RouterF(config-cmap)#match protocol http url "*root.exe*"*
RouterF(config-cmap)#exit

Next, we mark inbound Code Red traffic with a policy map using a specified DSCP value. In this example, we use a value of 1.

RouterF(config)#policy-map mark-inbound-http-hacks
RouterF(config-pmap)#class http-hacks
RouterF(config-pmap)#set ip dscp 1

Following that, we apply the service policy to the outside interface S0 so that inbound traffic will be marked.

RouterF(config)#int s0
RouterF(config-if)#service-policy input mark-inbound-http-hacks

Then, we define an access-list that will block any traffic marked with a dscp value of 1.

RouterF(config)#access-list 101 deny ip any any dscp 1 log
RouterF(config)#access-list 101 permit ip any any

Finally, the access list is applied outbound on the internal interface E0 where the web servers are located.

RouterF(config)#int e0
RouterF(config-if)#ip access-group 101 out

There are some limitations when using NBAR to block Code red attacks. Firstly, NBAR only recognizes the Code Red v1 and Code Red v2 URL request but not the Code-Red II URL request because Code-Red II spreads the GET request over multiple packets and NBAR only inspects the first packet. Secondly, if the web server is using the microsoft indexing service, the above method could block legitimate requests to the web server.

Performing router integrity checking

Given all the security features available in the Cisco IOS®, if the integrity of the router's configurations and IOS cannot be ensured, the features deployed will be futile. One solution is to build a simple tripwire-like system [Ref.19], to check the integrity of both the IOS running on the router and the latter's configurations. The solution is outlined as follow:

1) First, we need to obtain the baseline configurations (both running-config and startup-config) of the router and store it along with the image of the running IOS in a safe trusted central repository. The central repository could be some software version control tools like clearcase or CVS.

2) Next, we obtain the current configurations/IOS images of the router by tftp .

3) Following which, we can differ the configurations/IOS images using clearcase or CVS.

Step 2 and 3 should be performed periodically, to ensure that the router's configurations and IOS are not altered. However, there is one limitation with the solution outlined. The configurations are transfer in clear text over the network. To overcome this, IPSec could be used to encrypt the traffic.

Conclusion

In this paper, we have covered additional steps and features available in the Cisco IOS® that can be used to secure a network. The features discussed here are by no means an end to all the features available on the Cisco IOS®, interested readers are encouraged to find out other features from Cisco Connection Online.

By applying the security steps and features mentioned in this paper to secure the network, along with those in Degner's paper [Ref.1], the reader should be able to better secure the network. Even though, the configurations given here are for Cisco routers only, the reader could still apply some of the techniques on non-Cisco routers. For example, neighbor router authentication and route filtering.

References

1. Mark, Degner. "Securing Your Network With An Internet Access Router (or Getting Your Money's Worth From Your Cisco Gear)". April 4,2002.
URL: http://rr.sans.org/netdevices/cisco_gear.php (May 18,2002)

2. Cisco Systems. "Neighbor Router Authentication: Overview and Guidelines".
URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113ed_cr/secur_c/scprt5/scrouter.htm (May 19, 2002)

3. Cisco Systems. "Sample Configuration for Authentication in OSPF".
URL: http://www.cisco.com/warp/public/104/25.shtml (May 19, 2002)

4. Cisco Systems. "Improving Security on Cisco Routers". May 01, 2002
URL: http://www.cisco.com/warp/public/707/21.html (May 19,2002)

5. Cisco Systems. "Cisco Security Advisory: Multiple SSH Vulnerabilities". November 12,2001.
URL: http://www.cisco.com/warp/public/707/SSH-multiple-pub.html (May 20,2002)

6. Steven Acheson. "Secure Shell FAQ". February 16, 2001.
URL: http://www.employees.org/~satch/ssh/faq/ssh-faq-1.html#ss1.12 (May 20,2002)

7. The University of Texas at Austin. "A Detailed Review of Secure Shell". February 27, 2002. URL: http://www.tacc.utexas.edu/resources/user_guides/ssh_detailed/ (May 20,2002)

8. Cisco Systems. "IPSec". July 1,2002.
URL:
http://www.cisco.com/warp/public/cc/techno/protocol/ipsecur/ipsec/tech/ipsec_wp.htm (May 20, 2002)

9. Cisco Systems. "Configuring Internet Key Exchange Security Protocol". November 20, 2001.
URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/secur_c/scprt4/scike.htm (May 20, 2002)

10. Cisco Systems." Configuring IPSec Network Security". February 4,2002.
URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios121/121cgcr/secur_c/scprt4/scdipsec.htm#xtocid0 (May 20, 2002)

11. Microsoft Corporation. "Step-by-Step Guide to Internet Protocol Security (IPSec)". February 17, 2000. URL:
http://www.microsoft.com/technet/treeview/default.asp?url=/technet/prodtechnol/windows2000serv/howto/ispstep.asp (May 21,2002)

12. Chris Weber. "Using IPSec in Windows 2000 and XP". December 5, 2001.
URL: http://online.securityfocus.com/infocus/1519 (May 21, 2002)

13. Codex. "Cisco-ack-proof-concept". May 24, 2002.
URL: http://www.phate.net/docs/security/cisco-ack-proof-concept.txt (May 22, 2002)

14. Cisco Systems. "Cisco IOS® Software Release 11.3 New Features". June 30, 2002
URL: http://www.cisco.com/warp/public/cc/pd/iosw/iore/iore113/prodlit/706_pp.htm (May 22, 2002)

15. Cisco Systems. "Configuring IP Session Filtering (Reflexive Access Lists)".
URL:
http://www.cisco.com/univercd/cc/td/doc/product/software/ios113ed/113ed_cr/secur_c/scprt3/screflex.htm#xtocid87131 (May 23,2002)

16. Cisco Systems. "Context-based Access Control". July 27, 1999.

URL: http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/120newft/120t/120t5/iosfw2/iosfw2_2.htm#xtocid135950 (June 1,2002)

17. CERT. “CERT® Advisory CA-2001-19 "Code Red" Worm Exploiting Buffer Overflow In IIS Indexing Service DLL”. January 17, 2002. URL: http://www.cert.org/advisories/CA-2001-19.html (June 2, 2002)

18. Cisco Systems. “Using Network-Based Application Recognition and ACLs for Blocking the "Code Red" Worm”. May 15, 2002. URL: http://www.cisco.com/warp/public/63/nbar_acl_codered.shtml (June 2,2002)

19. Nicolas Fischbach & Sébastien Lacoste-Seris. “Protecting your IP Network Infrastructure”. Version 1.05BH. URL: http://www.securite.org/presentations/secip/ (June 4, 2002)

Last Updated: December 11th, 2017

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

SANS SEC460: Enterprise Threat Beta	San Diego, CAUS	Jan 08, 2018 - Jan 13, 2018	Live Event
SANS Security East 2018	New Orleans, LAUS	Jan 08, 2018 - Jan 13, 2018	Live Event
SANS Amsterdam January 2018	Amsterdam, NL	Jan 15, 2018 - Jan 20, 2018	Live Event
Northern VA Winter - Reston 2018	Reston, VAUS	Jan 15, 2018 - Jan 20, 2018	Live Event
SEC599: Defeat Advanced Adversaries	San Francisco, CAUS	Jan 15, 2018 - Jan 20, 2018	Live Event
SANS Dubai 2018	Dubai, AE	Jan 27, 2018 - Feb 01, 2018	Live Event
SANS Las Vegas 2018	Las Vegas, NVUS	Jan 28, 2018 - Feb 02, 2018	Live Event
SANS Miami 2018	Miami, FLUS	Jan 29, 2018 - Feb 03, 2018	Live Event
Cyber Threat Intelligence Summit & Training 2018	Bethesda, MDUS	Jan 29, 2018 - Feb 05, 2018	Live Event
SANS Scottsdale 2018	Scottsdale, AZUS	Feb 05, 2018 - Feb 10, 2018	Live Event
SANS London February 2018	London, GB	Feb 05, 2018 - Feb 10, 2018	Live Event
SANS Southern California- Anaheim 2018	Anaheim, CAUS	Feb 12, 2018 - Feb 17, 2018	Live Event
SANS Secure India 2018	Bangalore, IN	Feb 12, 2018 - Feb 17, 2018	Live Event
Cloud Security Summit & Training 2018	San Diego, CAUS	Feb 19, 2018 - Feb 26, 2018	Live Event
SANS Dallas 2018	Dallas, TXUS	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS Brussels February 2018	Brussels, BE	Feb 19, 2018 - Feb 24, 2018	Live Event
SANS Secure Japan 2018	Tokyo, JP	Feb 19, 2018 - Mar 03, 2018	Live Event
SANS New York City Winter 2018	New York, NYUS	Feb 26, 2018 - Mar 03, 2018	Live Event
CyberThreat Summit 2018	London, GB	Feb 27, 2018 - Feb 28, 2018	Live Event
SANS London March 2018	London, GB	Mar 05, 2018 - Mar 10, 2018	Live Event
SANS Cyber Defense Initiative 2017	OnlineDCUS	Dec 12, 2017 - Dec 19, 2017	Live Event
SANS OnDemand	Books & MP3s OnlyUS	Anytime	Self Paced