How to Configure the IOS HTTP Application Firewall

Configuring a new feature, the IOS HTTP Application Firewall, can further enhance the Cisco IOS Classic Firewall. HTTP uses TCP port 80 to transport Internet web services, which are commonly used on the network and rarely challenged with regard to their legitimacy and conformance to standards. Because traffic on TCP port 80 is typically allowed through the network without being challenged, many application developers are leveraging HTTP traffic as an alternative transport protocol in which to enable their application to travel through or even bypass the firewall. Most firewalls provide only packet-filtering capabilities that simply permit or deny HTTP without inspecting the data stream. An enabled Cisco IOS HTTP Application Firewall performs deep packet inspection to determine whether the packet is in compliance with the HTTP protocol. The IOS HTTP Application Firewall supports inspection for both HTTP and instant messaging protocols. In this configuration example, we configure the IOS HTTP Application Firewall to inspect HTTP traffic to look for and report on the use of HTTP tunneling applications such as peer-to-peer (P2P) networks and various instant messenger clients that tunnel over TCP port 80 (HTTP). This configuration will be applied to the IOS Classic Firewall configuration that we configured earlier in this section. The configuration can be broken down into two steps. These are in addition to the three steps required for the IOS Classic Firewall that we covered earlier: STEP 1 Define the HTTP policy. STEP 2 Apply the HTTP policy to an inspection rule. Step 1: Define the HTTP Policy The first step is to define the HTTP policy that we will be applying to the firewall configuration. In this example, we define standard HTTP policy that will allow nonstandard use of the HTTP protocol, but it will alarm to a configured logging source on the router. We need to be in the Application Firewall configuration mode to enter this configuration. This is designated as cfg-appfw-policy: Router(config)# **appfw policy-name MYAPPFW** Router(cfg-appfw-policy)# **application http** Router(cfg-appfw-policy-http)# **strict-http action allow alarm** Router(cfg-appfw-policy-http)# **content-length maximum 1 action allow alarm** Router(cfg-appfw-policy-http)# **content-type-verification match-req-rsp action allow alarm** Router(cfg-appfw-policy-http)# **max-header-length request 1 response 1 action allow alarm** Router(cfg-appfw-policy-http)# **max-uri-length 1 action allow alarm** Router(cfg-appfw-policy-http)# **port-misuse default action allow alarm** Router(cfg-appfw-policy-http)# **request-method rfc put action allow alarm** Router(cfg-appfw-policy-http)# **transfer-encoding type default action allow alarm** The first configuration line creates an HTTP policy called MYAPPFW. We then set this to perform HTTP inspection with the **application http** command. We are then in cfg-appfw-policy-http configuration mode. The following commands are then entered: · **strict-http action allow alarm** This command allows nonconforming HTTP packets and creates an alarm. · **content-length maximum 1 action allow alarm** This command allows traffic that exceeds the maximum content length for HTTP and creates an alarm. · **content-type-verification match-req-rsp action allow alarm** This command allows traffic that exceeds the configured policy and creates an alarm. · **max-header-length request 1 response 1 action allow alarm** This command allows traffic that exceeds the configured policy and creates an alarm. · **max-uri-length 1 action allow alarm** This command allows traffic that exceeds the configured policy and creates an alarm. · **port-misuse default action allow alarm** This command enables the inspection of any application that misuses the HTTP protocol. This is typical for HTTP tunneling applications and instant messenger applications. · **request-method rfc put action allow alarm** This command states that HTTP requests will be inspected for RFC compliance and for Remote Copy Protocol (RCP) extension methods. · **transfer-encoding type default action allow alarm** This command allows the inspection of all transfer encoding types. This policy that we have just presented configures the HTTP inspection engine to allow the matching traffic and send a message to the configured syslog server. The policy does not block anything, it just reports on the items that are matched in the policy. Step 2: Apply the HTTP Policy to an Inspection Rule After the HTTP policy has been created, the next step is to apply the HTTP policy to an inspection rule. In the Classic IOS Firewall configuration example, we configured an inspection rule called MYFW. We are now going to add the MYAPPFW HTTP policy to the MYFW inspection rule. Before we do this, we also have to enable HTTP inspection on the inspection rule: Router(config)# **ip inspect name MYFW http** Router(config)# **ip inspect name MYFW appfw MYAPPFW** The first line enables HTTP inspection, and the second line links the HTTP policy to the inspection rule.