

Securing Wireless LAN Controllers (WLCs)

Document ID: 109669

Introduction

Prerequisites

Requirements

Components Used

Conventions

Traffic Handling in WLCs

Controlling Traffic

Controlling Management Access

CPU ACLs

Example

Testing Before CPU ACL

Testing After the CPU ACL

Strict CPU ACLs

Control Plane Policing

Strong Encryption for HTTPs traffic

Session Control

Telnet/SSH Settings

Console Port

Putting all Together

Security Practices

Related Information

Introduction

This document offers an overview of several important aspects needed to handle the security interaction between Wireless LAN Controllers (WLCs) and the network where they are connected. This document focuses primarily on traffic control, and does not address WLANs security policies, AAA or WPS.

Topics affecting the traffic with destination to the controller are covered in this document, and not related to traffic which is related to user to network .

Note: Validate changes before applying them to your network, as some of the examples in this document can block administrative access to your controllers if applied incorrectly.

Prerequisites

Requirements

Cisco recommends that you have knowledge of these topics:

- Knowledge of how to configure the WLC and Lightweight Access Point (LAP) for basic operation
- Basic knowledge of the OSI model
- Understanding how Access Control List (ACL) works

Components Used

The information in this document is based on these software and hardware versions:

- Cisco 2000 / 2100 / 4400 Series WLC that runs firmware 4.2.130.0, 5.2.157.0 or later

The information in this document was created from the devices in a specific lab environment. All of the devices used in this document started with a cleared (default) configuration. If your network is live, make sure that you understand the potential impact of any command.

Conventions

Refer to Cisco Technical Tips Conventions for more information on document conventions.

Traffic Handling in WLCs

One critical component on network security is traffic control. On any deployment, it is very important to block types of traffic arriving at devices in order to prevent potential security issues (DoS, information loss, privilege escalation, etc).

On the WLC, traffic control is affected by an important fact: there are two components handling traffic in the device:

- CPU Main processor that takes care of all management activity, RRM, LWAPP control, authentication, DHCP, etc.
- NPU Network processor that takes care of fast traffic forwarding for authenticated clients (wired to wireless and vice versa).

This architecture allows a fast traffic forwarding, and reduces the load on the main CPU, which can then dedicate all its resources for high level tasks.

This architecture is found on the 4400, WiSM and 3750 integrated controllers. For 2106 and NM-WLC and related controllers, the forwarding is done in software, also by the main CPU. Therefore, it does take a higher tax on the CPU. That is why these platforms offer a lower user and AP count support.

Controlling Traffic

Anytime you want to filter traffic in relation to a WLC, it is important to know if this is a user to network traffic or it is towards the main CPU.

- For any traffic to the CPU, for example, management protocols such as SNMP, HTTPS, SSH, Telnet, or network services protocols such as Radius or DHCP, use a CPU ACL is used.
- For any traffic to and from a wireless client, including traffic going through a EoIP tunnel (guest access), an Interface ACL, a WLAN ACL, or a per user ACL is used.

Traffic is defined to the CPU, as traffic which is entering the controller, with destination to the management IP address, any of the dynamic interfaces or the service port address. AP-Manager does not handle any other traffic except LWAPP/APWAP.

Controlling Management Access

WLCs have a session level access control for management protocols. It is important to understand how they work in order to prevent incorrect assessment on what is allowed or not allowed by the controller.

The commands to restrict what management protocols are allowed are (on a global scope):

- **config network ssh enable|disable** This enables or disables the SSH service on the controller. This is enabled by default. Once disabled, the port (TCP 22) will not be reachable.
- **config network telnet enable|disable** This enables or disables the telnet service on the controller. This is disabled by default. Once disabled, the port (TCP 23) will not be reachable.
- **config network http enable|disable** This enables or disables the http service on the controller. The port (TCP 80) is not longer reachable. This is disabled by default.
- **config network https enable|disable** This enables or disables the https service on the controller. This is enabled by default. Once disabled, the port (TCP 443) will not be reachable.
- **config snmp version v1|v2|v3 enable|disable** This enables or disables specific versions of SNMP service on the controller. You need to disable all to prevent SNMP access to controller, unless using an ACL.
- **config network mgmt-via-wireless enable|disable** This prevents that clients associated to this controller can access management protocols to it (ssh, https, etc). This does not prevent or close the TCP corresponding ports from the point of view of the wireless device. This means that a wireless device, when this is set to disable, can open an SSH connection, if the protocol is enabled. However, the user will never be presented with a logging prompt.
- **config network mgmt-via-dynamic-interface enable|disable** This prevents that devices on the same VLAN as the controller can access management protocols to it (ssh, https, etc) to the corresponding dynamic interface address on that VLAN. This does not prevent or close the TCP corresponding ports from the point of view of the device. This means that a device, when this is set to disable, can open an SSH connection, if the protocol is enabled. However, the user will never be presented with a logging prompt. Additionally, the management address will always remain accessible from a dynamic interface VLAN, unless a CPU ACL is on place.

For example, this is the configuration using the above information:

```
(Cisco Controller) >show network summary

RF-Network Name..... 4400
Web Mode..... Disable
Secure Web Mode..... Enable
Secure Web Mode Cipher-Option High..... Disable
Secure Web Mode Cipher-Option SSLv2..... Enable
Secure Shell (ssh)..... Enable
Telnet..... Disable
Ethernet Multicast Mode..... Enable   Mode: Ucast
Ethernet Broadcast Mode..... Disable
AP Multicast Mode..... Unicast
IGMP snooping..... Disabled
IGMP timeout..... 60 seconds
User Idle Timeout..... 300 seconds
ARP Idle Timeout..... 300 seconds
Cisco AP Default Master..... Disable
AP Join Priority..... Disable
Mgmt Via Wireless Interface..... Disable
Mgmt Via Dynamic Interface..... Disable
Bridge MAC filter Config..... Enable
Bridge Security Mode..... EAP
Mesh Full Sector DFS..... Enable
Over The Air Provisioning of AP's..... Enable
```

```

Apple Talk ..... Disable
AP Fallback ..... Enable
Web Auth Redirect Ports ..... 80
Fast SSID Change ..... Disabled
802.3 Bridging ..... Disable
IP/MAC Addr Binding Check ..... Enabled

```

```
(Cisco Controller) >show acl cpu
```

```

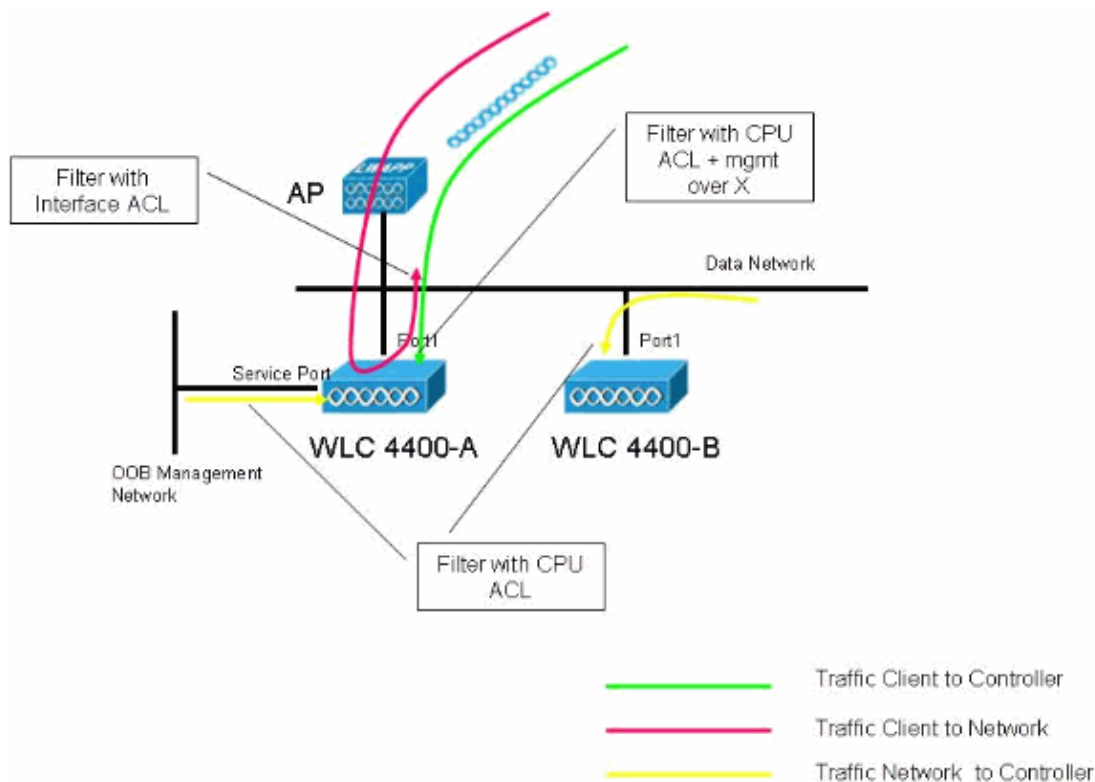
CPU Acl Name..... NOT CONFIGURED
Wireless Traffic..... Disabled
Wired Traffic..... Disabled

```

You can conclude that:

- Telnet and HTTP will not be available, so all interactive management traffic to the controller will be done through HTTPS/SSH (encrypted).
- A wireless user associated to this controller will not be able to get administrative access.
- If a wireless user, associated to this controller, does a port scan, it will show SSH and HTTP as open, even though no administrative access is allowed.
- If a wired user (same VLAN as a dynamic interface) does a port scan, it will show SSH and HTTP as open, even though no administrative access is allowed.

It is important to note that in environments with more than one controller on the same mobility group, the relationship of what is a wireless client is only to the currently associated controller. Therefore, if one client is associated to controller A, then for a controller B on the same mobility group, this client is a device coming from a VLAN/dynamic interface. This is important to take into account on Management over wireless setting. See this diagram for an example of where to put a traffic restriction, and what commands can affect each ingress point:



CPU ACLs

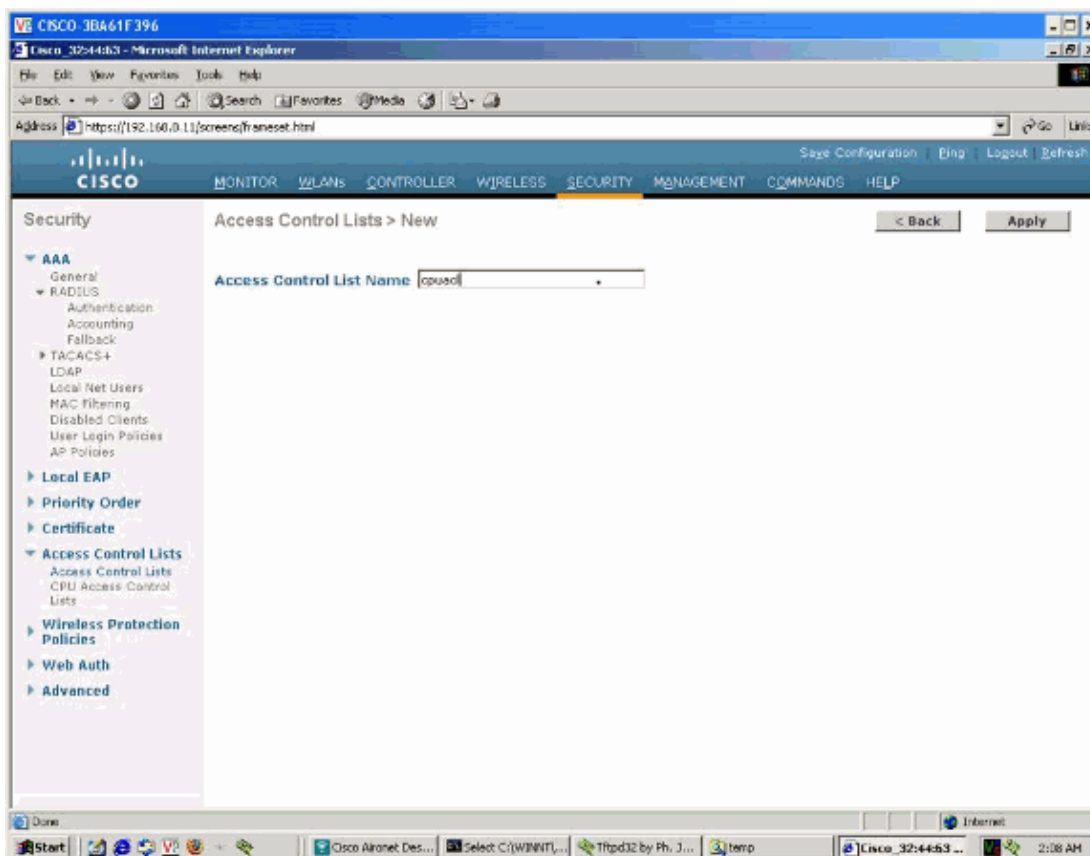
Whenever you want to control which devices can talk to the main CPU, a CPU ACL is used. It is important to mention several characteristics for these:

- CPU ACLs only filter traffic towards the CPU, and not any traffic exiting or generated by the CPU. The direction field on a CPU ACL should always be Inbound or Any.
- Full support for CPU ACLs for all controller IP management and dynamic addresses is only present on 4.2.130.0 and later.
- CPU ACLs blocking service port traffic is only present in 5.0 and later.
- CPU ACLs can be set to block only for wired traffic (incoming from connected vlans), wireless (incoming from the associated clients) or both.
- When a CPU ACL is designed, it is important to allow control traffic between controllers. The **sh rules** command can offer a quick view of traffic permitted to CPU ACL on normal conditions.
- The controller has a set of filtering rules for internal processes, which can be checked with the **sh rules** command. ACLs do not affect these rules, nor can these rules be modified on the fly. CPU ACL takes precedence over them.
- LWAPP or CAPWAP data traffic is not affected by CPU ACLs rules on 4400 based controllers, control traffic is affected (if doing an strict ACL, you need to explicitly permit it).

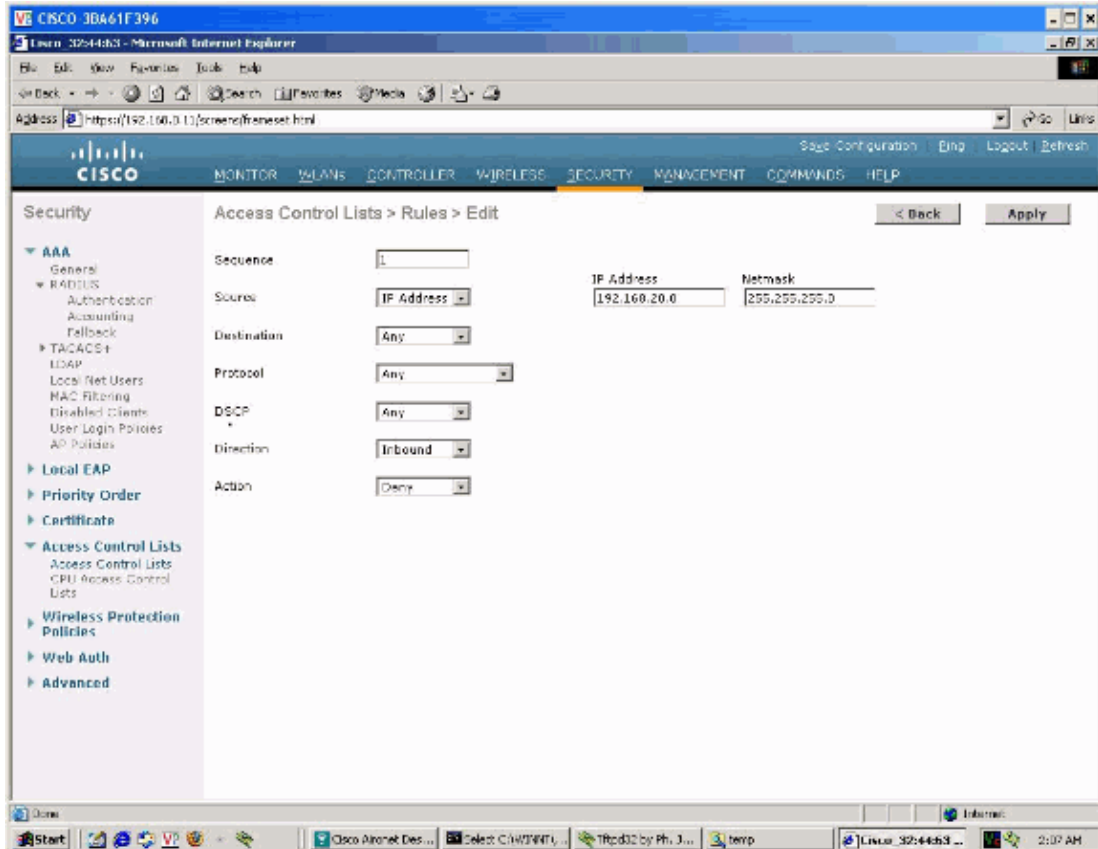
Example

For example, you might want to block all traffic coming from the dynamic interface/VLAN (192.168.20.0/24) where users are associated, towards the CPU, but any other traffic is allowed. This should not prevent wireless clients to get a DHCP negotiated address.

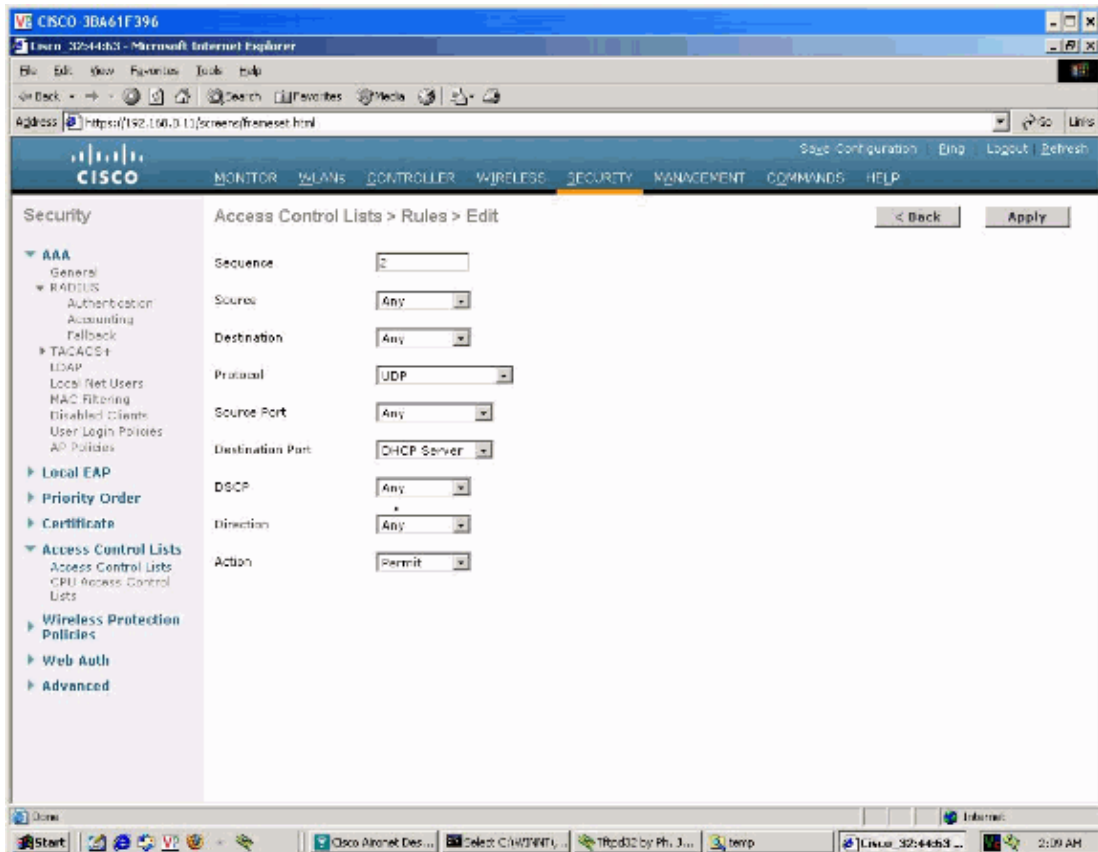
1. As first step, an access list is created:



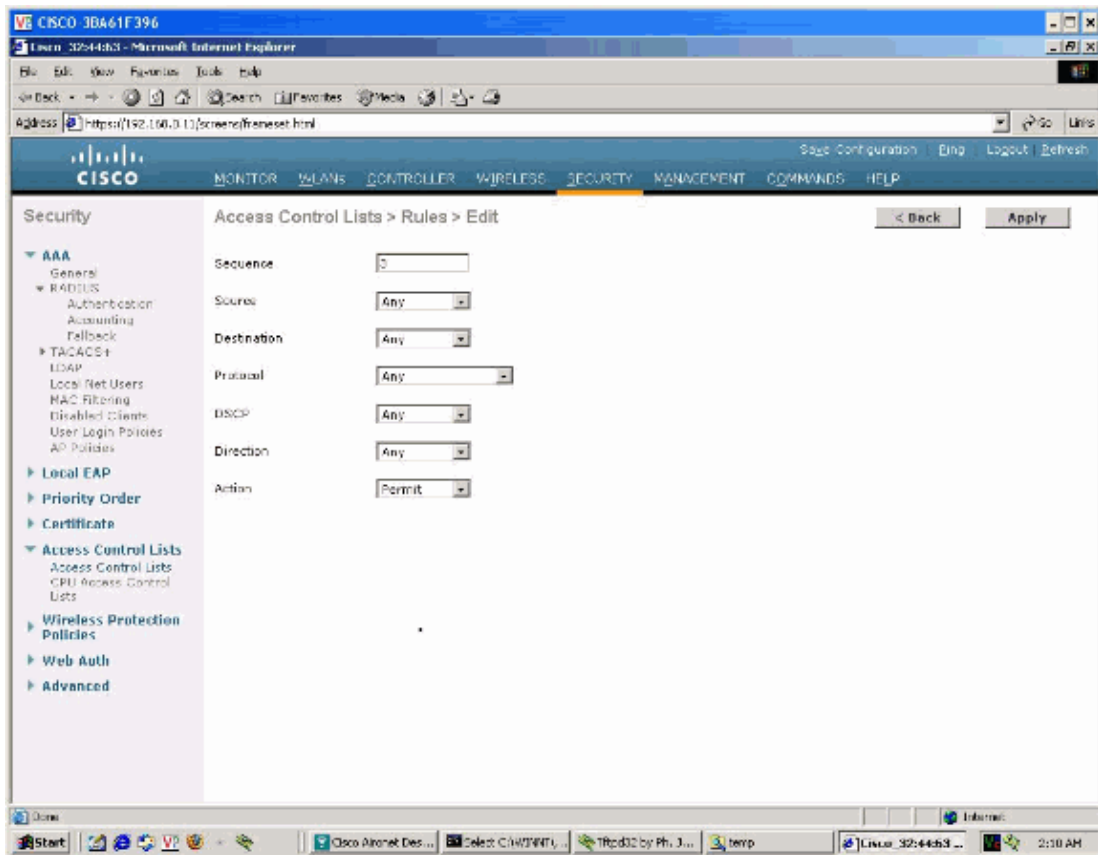
2. Click **Add new rule**, and set it to block all source traffic coming from 192.168.20.0/24 to any destination.



3. Add a second rule, for DHCP traffic, with destination server port, but with permit action:

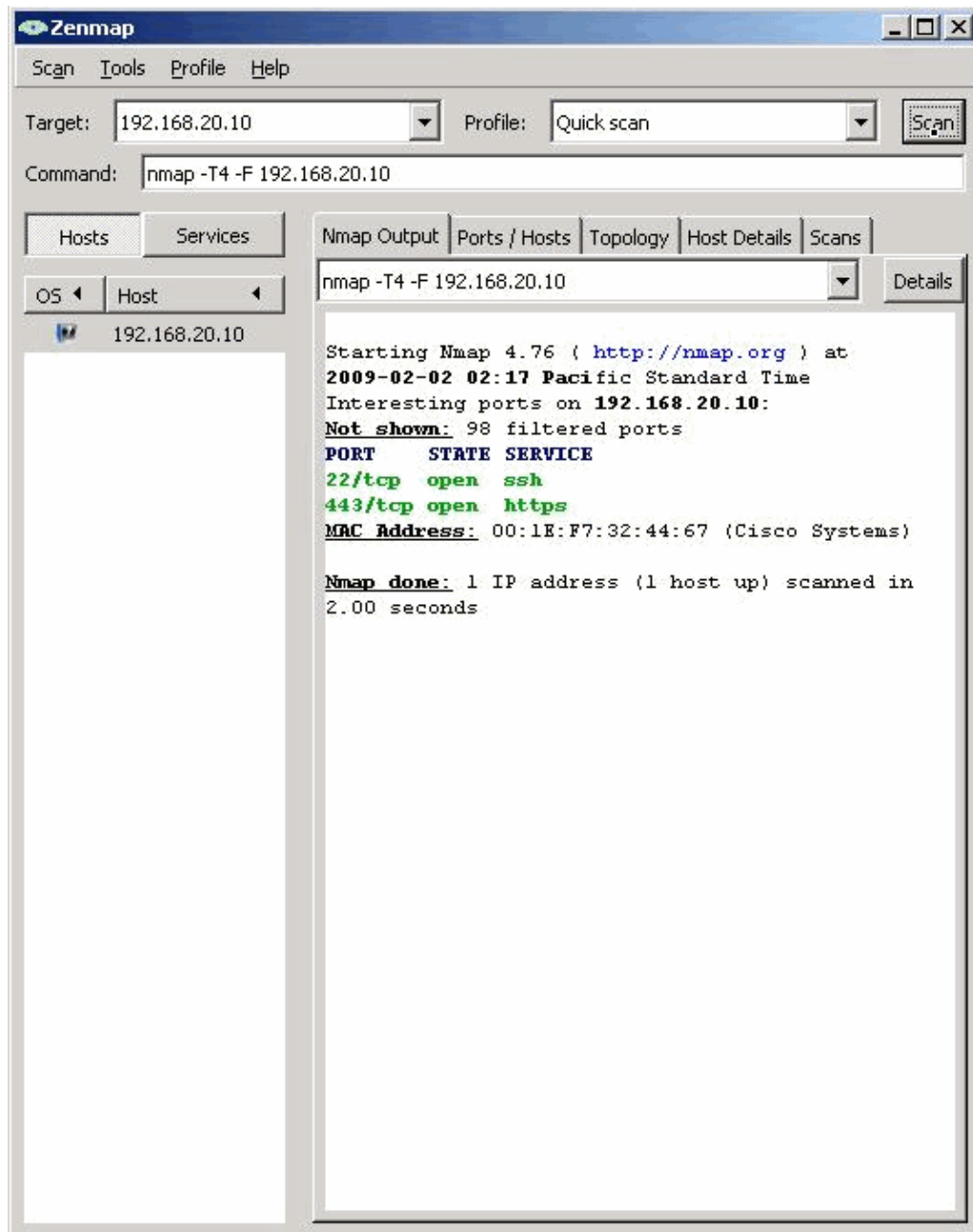


Then, per company security policies, all other traffic is allowed:



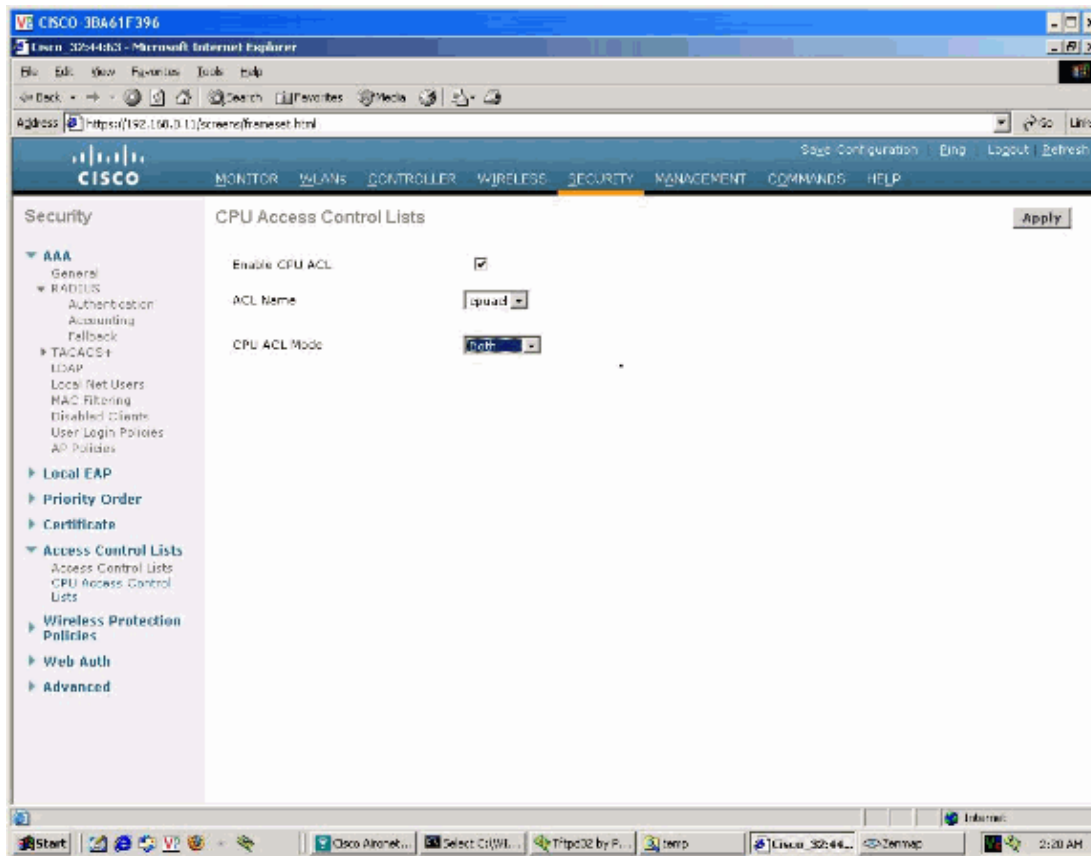
Testing Before CPU ACL

In order to validate the effect of the CPU ACL, you can perform a quick scan from an associated wireless client on RUN Status in order to see the current open ports, based on the configuration, before applying the CPU ACL:

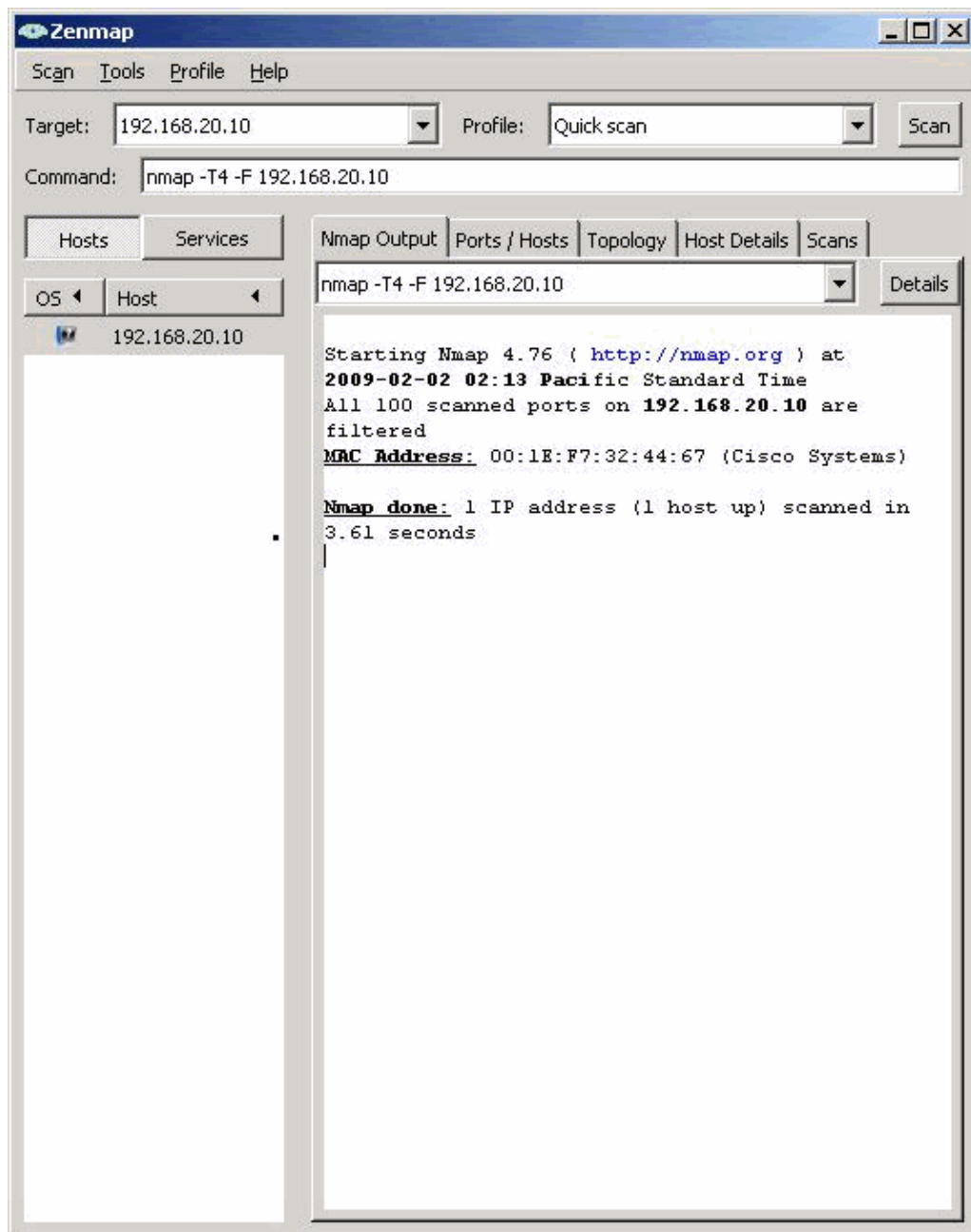


Testing After the CPU ACL

Go to **Security > Management > CPU Access Control List**. Click **Enable CPU ACL**, and select the ACL that was previously created. Then, choose **Both** as direction in order to insure this is applied to traffic from wireless clients, and from other devices on Dynamic Interface VLANs:



Now, if the same scan used before is repeated, all ports of the controller are shown as closed:



Strict CPU ACLs

If the security policies demand a `deny any` as last line for a policy, it is important to understand that there are several types of traffic sent between controller on the same mobility group for RRM, mobility and other tasks, and that you might have traffic proxied by the controller to itself for some operations, in particular DHCP, where controller on DHCP proxy mode (the default) can generate traffic to itself with destination UDP 1067 for processing.

For a complete list of ports allowed by the internal default forwarding rules, check the output of the `sh rules` command. The complete list analysis is beyond the scope of this document.

You can check which ACL rules are being hit by traffic with the `config acl counter start` command. The counters can be displayed with the `sh acl detail ACLNAME` command.

Control Plane Policing

One aspect of protecting a network device, is to make sure that it is not overwhelmed with more management traffic that it can process. On all controllers, after 4.1 code, there is a control plane limiting enabled by default, which will kick in if traffic for CPU exceeds the 2 mbps.

On busy networks, it is possible to observe the limiting in effect (for example, dropped monitor pings to CPU). The feature can be controlled with the **config advanced rate** command. You can only enable or disable it, but not set rates or against which traffic it will act first.

On normal operations, it is recommended this is left enabled.

Strong Encryption for HTTPs traffic

By default, the controller offers both high and low strength ciphers to insure compatibility with older browsers during HTTPS setup. The controller has available from 40 bits RC4, 56 Bits DES, up to AES 256 Bits. The selection of the strongest cipher is done by the browser.

In order to make sure that only strong ciphers are used, you can enable them with the **config network secureweb cipher-option high enable** command, so only 168 3DES or 128 AES and higher cipher lengths are offered by the controller on HTTPS management access.

Session Control

Telnet/SSH Settings

By default, the controller allows a maximum of 5 concurrent users, with a timeout of 5 minutes. It is critical that these values are configured adequately in your environment, as setting them to unlimited (zero) can open the door to potential denial of service against controllers, if users were to try a brute force attack against them. This is an example of default settings:

```
(Cisco Controller) >show sessions

CLI Login Timeout (minutes)..... 5
Maximum Number of CLI Sessions..... 5
```

Remember that by design, even if management over wireless or dynamic interface is disabled, a device can still make an SSH connection to the controller. This is a CPU taxing task, and WLC limits the number of simultaneous sessions, and for how long using these parameters.

The values can be adjusted with the **config sessions** command.

Console Port

The serial port has a separated timeout value, which is set to 5 minutes by default, but it is commonly changed to 0 (unlimited) during troubleshooting sessions.

```
Cisco Controller) >show serial

Serial Port Login Timeout (minutes)..... 5
Baud Rate..... 9600
Character Size..... 8
Flow Control:..... Disable
Stop Bits..... 1
```

Parity Type:..... none

It is advisable to use the default of 5 minutes. This prevents anyone having physical access to the controller to gain administrative access, in case a logged in user on the console port leaves the session open. The values can be adjusted with the **config serial** command.

Putting all Together

After checking the different aspect of securing a WLC, this can be summarized:

- It is important to prevent devices other than indented management stations to access WLC, not only disabling non-used protocols, but also by limiting access on layer 4/layer 3 with CPU ACLs.
- Rate limiting should be enabled (it is by default).
- Controlling access through **management over X** commands is not enough for secure installations, as users can still access management protocols talking directly to the management IP address, using CPU and memory resources.

Security Practices

Here are some of the security practices:

- Create CPU ACL dropping access from all dynamic interface VLANs or subnetworks. However, allow DHCP traffic to server port (67) so clients can obtain DHCP negotiated address if DHCP proxy is enabled (it is by default). If the dynamic interface has a public IP address, it is recommended to have ACL rule denying all traffic from unknown sources to the dynamic interface address.
- Set all ACL rules as inbound or with direction **any**, and mark them as applied as **both** (wired and wireless option).

How to validate:

```
(Cisco Controller) >show acl cpu

CPU Acl Name..... acl1
Wireless Traffic..... Enabled
Wired Traffic..... Enabled
```

- Enable control plane limiting (it is enabled by default).

How to validate:

```
(Cisco Controller) >show advanced rate

Control Path Rate Limiting..... Enabled
```

- Always use encrypted management protocols (HTTPS, SSH). This is the default configuration for interactive management. For SNMP you might need to enable V3 to allow encrypted/authenticated SNMP traffic. Remember to reload controller if you make changes to SNMP configuration.

This is how to validate:

```
(Cisco Controller) >show network summary

RF-Network Name..... 4400
Web Mode..... Disable
Secure Web Mode..... Enable
Secure Web Mode Cipher-Option High..... Enable
Secure Web Mode Cipher-Option SSLv2..... Enable
```

```
Secure Shell (ssh)..... Enable
Telnet..... Disable
...
```

- Enable high encryption for HTTPS (this is disabled by default).
- It is a good idea to set up a validated server certificate for HTTPS access to your controller (signed by your trusted CA), replacing the self signed certificate installed by default.
- Set session and console timeout to 5 minutes.

```
(Cisco Controller) >show serial
```

```
Serial Port Login Timeout (minutes)..... 5
Baud Rate..... 9600
Character Size..... 8
Flow Control:..... Disable
Stop Bits..... 1
Parity Type:..... none
```

```
(Cisco Controller) >show sessions
```

```
CLI Login Timeout (minutes)..... 5
Maximum Number of CLI Sessions..... 5
```

Related Information

- [Lightweight Access Point FAQ](#)
- [Wireless LAN Controller \(WLC\) Troubleshoot FAQ](#)
- [Cisco Wireless LAN Controller Module Q&A](#)
- [Radio Resource Management under Unified Wireless Networks](#)
- [Technical Support & Documentation – Cisco Systems](#)

[Contacts & Feedback](#) | [Help](#) | [Site Map](#)

© 2008 – 2009 Cisco Systems, Inc. All rights reserved. [Terms & Conditions](#) | [Privacy Statement](#) | [Cookie Policy](#) | [Trademarks of Cisco Systems, Inc.](#)

Updated: Feb 25, 2009

Document ID: 109669
