aruba
a Hewlett Packard Enterprise company

THREAT LABS

ARUBA THREAT LABS

ARUBAOS HARDENING GUIDE

OCTOBER, 2020

CONTENTS

REVISION HISTORY

Name	Date	Changes
Jon Green	2-July-2014	Initial public release
Jon Green	30-October-2015	Updated Common False Positives and other misc. edits
Jon Green	9-December-2016	Added additional common false positives
Jon Green	7-December-2017	Added additional false positives
Jon Green	3-March-2018	Updated section on SSH to include new configuration commands
Jon Green	25-May-2018	Added section on password recovery/console port
Rick Farina	2-August-2018	Added some new open port information
Jon Green	14-February-2019	Additional ports and false positives
Jon Green	28-October-2020	Additional ports and false positives, AOS 8.x changes, AP console port

Introduction

This document has been produced to assist Aruba customers and partners in configuring Aruba mobility controllers, access points, and switches in the most secure manner. It should be noted that security recommendations often involve tradeoffs; not every recommendation in this document will be appropriate for every situation. In general, however, recommendations in this document represent security best practices and should be followed wherever network security is a priority.

External Security Testing and Accreditation

Aruba Networks spends a significant amount of time and money conducting independent third-party security testing of its products. While the majority of this testing is relevant to – and required by – government agencies, it has value to all types of users. In some cases, organizations may choose to rely on recognized security testing authorities rather than conducting their own product testing.

- **FIPS 140-2**

  The Federal Information Processing Standard 140-2 is a system for testing and certifying cryptographic modules. As part of this testing, a laboratory accredited by the US and Canadian governments examines design documentation, source code, and development practices, in addition to conducting extensive testing of cryptographic functions. Products that implement FIPS 140-2 validated cryptography are assured to be using cryptography correctly. Note that only FIPS software variants of ArubaOS (found in the ArubaOS-FIPS folder on the support website) are actually FIPS validated. However, non-FIPS software images are built from the same source code base, with the major difference being a number of self-tests which are run in FIPS software and are not run in non-FIPS software. More information about FIPS 140-2 may be found at http://csrc.nist.gov/groups/STM/cmvp/standards.html.

  All Aruba mobility controllers have received FIPS 140-2 validation. Most access points have also been validated. Mobility access switches have not yet been validated, but are currently in-process. A complete list of validated modules may be found at http://csrc.nist.gov/groups/STM/cmvp/documents/140-1/140val-all.htm.

- **Common Criteria**

  The Common Criteria for Information Technology Security Evaluation (abbreviated as CC) is an international standard (ISO/IEC 15408) for computer security evaluation. It is recognized by the governments of approximately 47 nations, and products that have received CC certificates are generally accredited for use in unclassified government systems. Whereas FIPS 140-2 is focused on cryptography, Common Criteria is focused on “everything else” that is security relevant, including management protocols, authentication mechanisms, vulnerabilities, and selections of parameters such as cipher suites. In the United States, Common Criteria is administered and controlled by NIAP – the National Information Assurance Partnership. Information about NIAP and Common Criteria may be found at https://www.niap-ccevs.org.

  All Aruba mobility controllers have completed Common Criteria evaluation under the Network Device Protection Profile (NDPP), the VPN Gateway extension profile (VPNGW) and Stateful Traffic Filter Firewall (TTFW) extension profile. All mobility controllers and most Aruba access points have also completed evaluation under the Wireless LAN Access System Protection Profile. Information about evaluated products may be found at https://www.commoncriteriaportal.org/products/.

- **US Department of Defense**

  Aruba products are widely deployed within the US Department of Defense (DoD). In addition to requiring validation under FIPS 140-2 and Common Criteria, the DoD also imposes additional testing related to interoperability and information assurance. Successful completion of this testing leads to

a product being listed on the Unified Capabilities Approved Product List (UC-APL) which can be viewed at https://aplits.disa.mil. Aruba mobility controllers and access points have been listed on the UC-APL multiple times, with Aruba repeating this testing approximately every 12-18 months.

Internal Security Testing

Each ArubaOS release goes through extensive quality assurance testing. As part of the testing process, several commercial vulnerability scanners are used. These include:

- QualysGuard
- nCircle
- Nessus
- Retina

Any findings returned by these scanners are examined to determine if they are genuine vulnerabilities or false positives. Actual vulnerabilities will cause a bug to be opened.

In addition to quality assurance testing, an internal group known as Aruba Threat Labs provides advanced vulnerability research against Aruba products. Aruba Threat Labs conducts penetration testing through both black-box and white-box testing, also including source code analysis. From time to time, Aruba Threat Labs also contracts with external third-party penetration testing firms to conduct targeted testing.

Vulnerability Management Process

Aruba publishes a vulnerability response policy at http://www.arubanetworks.com/support-services/security-bulletins/. This location also hosts security advisories published by Aruba. An RSS feed is available from this page as well. Any customer with an active support contract will receive vulnerability advisories by email; interested parties who want to receive advisories but who do not have a support contract can subscribe to the thread at http://community.arubanetworks.com/t5/AAA-NAC-Guest-Access-BYOD/Security-vulnerability-advisories/m-p/176738 to be notified when an update is posted.

Bug Bounty Program

Aruba operates a bug bounty program, through which security researchers are paid a reward for finding and reporting security vulnerabilities in Aruba products. The program is managed by BugCrowd, a third-party company that manages the researcher pool, reward payout, and the tracking and reporting process on behalf of Aruba. For more information, see http://www.bugcrowd.com.

Typical Vulnerability Scan Results

It is extremely common for customers to run their own vulnerability scans against Aruba devices. This section documents common results and answers frequently asked questions.

Open Ports

The following table lists all ports that are used by a device running ArubaOS. The table includes notes on what the port is used for, and whether or not it may be blocked using firewall rules. See the ArubaOS User Guide under the heading “Understanding Default Open Ports” for up-to-date information for specific versions of ArubaOS.

Port Number	Explanation	Notes
17/TCP	Used to support Nortel Contivity VPN clients	Can safely be blocked for most users. Vulnerability scanners report this as “quote of the day” and may flag an alert for “quote of the day traffic amplification”.
21/TCP	FTP. Used by Aruba APs to perform software image downloads from the controller.	FTP server supports download-only of AP image files. For this reason, security of this port is not a high priority (i.e. unsecure authentication methods may be used). Most APs prior to the AP225 shipped from the factory without an ArubaOS software image. FTP is required for these APs to boot the first time. APs which are configured for CPsec will perform software image downloads inside an IPsec tunnel; these APs do not require access to this port. Once a network is operating entirely in CPsec mode, this port may be safely blocked.
22/TCP	SSH. Used for administrative access to the ArubaOS command line.	It is recommended to enable access to this port only from trusted subnets.
23/TCP	Telnet. Used for administrative access to the ArubaOS command line.	Port is open by default, but connection attempts will be immediately closed. Telnet can be enabled through the configuration command “telnet cli”. Aruba does not recommend enabling telnet. This port may be blocked by firewall rules if telnet is not needed. This port is not open, and cannot be enabled, in the FIPS version of ArubaOS.
53/UDP	DNS responder. This service responds to all DNS queries with the IP address of the mobility controller. Often used to help APs boot when attached to isolated networks without an operating DNS server.	May be blocked by firewall rules without any loss in service. Vulnerability scanners often report problems with this port, such as “DNS cache snooping”. Because this port does not connect to an actual DNS server, warnings may be safely ignored.
67/UDP	DHCP server. ArubaOS is capable of providing DHCP server functionality when configured to do so.	Will be disabled automatically if DHCP service is not enabled.

80/TCP	HTTP. Accepts connections for both Captive Portal and for WebUI administrative management. Redirects to other ports using HTTPS.	May be blocked if not needed.
123/UDP	NTP. Provides time synchronization service to APs.	APs configured for CPsec will run NTP inside an IPsec tunnel; these APs do not require access to this port. Once a network is operating entirely in CPsec mode, this port may be safely blocked.
161/UDP	SNMP. Provides SNMP management access.	Disabled by default – enabled when a SNMP community is configured. Access to this port should be restricted to authorized SNMP management systems, such as AirWave. Aruba recommends the use of SNMPv3.
443/TCP	HTTPS. Used for captive portal authentication, WebUI administrative management, and VIA.	WebUI will redirect to TCP/4343 unless “web-server web-https-port-443” has been configured. Captive portal will redirect to TCP/8081. VIA clients require this port to be available in order to perform profile download, and for SSL fallback mode.
500/UDP	ISAKMP/IKE. Used by IPsec.	May be blocked if the controller is not being used as a VPN server. VIA, RAPs, and APs operating in CPsec mode will use UDP/4500 by default to establish their IPsec tunnels.
514/UDP	Syslog. The controller operates as a syslog receiver for log messages from APs.	APs operating in CPsec mode will use the syslog protocol inside an IPsec tunnel; these APs do not require access to this port. This port may be blocked if not needed.
1701/UDP	L2TP. Used for VPN termination.	Port may be safely blocked if the controller is not being used as a VPN server for L2TP.
1723/TCP	PPTP. Used for VPN termination.	Port may be safely blocked if the controller is not being used as a VPN server for PPTP.
4343/TCP	HTTPS. Used for WebUI administrative management.	Connections to TCP/443 will redirect to this port by default, unless WebUI access through TCP/443 has been enabled.
4500/UDP	ISAKMP/IKE NAT Traversal. Used by VIA client, RAPs, and APs operating in CPsec mode.	This port should not be blocked.
6633/TCP	OpenFlow. Used for TLS connections between OpenFlow controllers and OpenFlow agents.	If this port is blocked, AirGroup and UCC will not function.

8080/TCP	HTTP. Used for captive portal.	May be blocked if captive portal is not in use.
8081/TCP	HTTPS. Used for captive portal.	May be blocked if captive portal is not in use.
8082/TCP	Used for Single Sign-On with other Aruba infrastructure	May be blocked if single sign-on is not in use.
8085/TCP	Enabled by configuration. Used for VIA to perform profile download when certificate-based VIA authentication is enabled.	May be blocked if not needed.
8088/TCP	HTTP/HTTPS. Used for captive portal for proxied clients.	May be blocked if captive portal is not in use with clients that require a web proxy.
9070-9080	Ports 9070 to 9080 are opened by OpenFlow Controller to be used by registered apps.	If these ports are blocked, AirGroup and UCC will not function.
9199/TCP	Data replication between MDs, and between MD and MM.	This port should not be blocked if Mobility Master is in use. Note: Need for this port being open is under review. Normally this traffic flows inside a CPsec tunnel.
32000/TCP	Used for integration with Microsoft Lync/Skype for Business	May be blocked if Lync/Skype for Business integration is not being done.
15260/TCP	MM connects to MD on this port for additional data collection that is not available through other messaging protocols.	Do not block this port if Mobility Master is in use. Note: Need for this port being open is under review. Normally this traffic flows inside a CPsec tunnel.

Common False Positives

The most common type of false positive seen by vulnerability scanners occurs when the scanner looks only at a version number presented as part of a protocol handshake. For example, a scan against a controller's SSH service may indicate that the SSH server is OpenSSH version 5.8. If the scanning tool's database finds known vulnerabilities for OpenSSH 5.8, it will report that the controller is vulnerable. Most vulnerability scanners *do not actually attempt to exploit vulnerabilities*, so the resulting report should be viewed as a list of *possible* vulnerabilities. Aruba incorporates a number of open-source packages within ArubaOS, such as Apache and OpenSSH. In the interest of software stability, Aruba typically does *not* update open-source packages to their latest version when a security vulnerability is found. This is because, in addition to security fixes, there may be potentially thousands of other source code changes which may introduce bugs. Instead, Aruba will patch specific vulnerabilities by fixing only the flaw itself.

The following table includes a number of common false positives for ArubaOS:

Service	Reference	Notes
OpenSSH	CVE-2016-10009 CVE-2016-10010 CVE-2016-10011 CVE-2016-10012 CVE-2016-6210 CVE-2016-0778 CVE-2016-0777 CVE-2015-6564 CVE-2015-6563 CVE-2015-5600 CVE-2015-5352 CVE-2014-1692 CVE-2014-2532 CVE-2014-2653 CVE-2011-5000 CVE-2011-4327 CVE-2010-5107 CVE-2010-4755 CVE-2008-3259 CVE-2007-4752 CVE-2007-2243 CVE-2007-2768 CVE-2004-1653	These vulnerabilities were either patched in source code, or are non-applicable for ArubaOS due to configuration or limited use of a particular feature. No currently supported version of ArubaOS contains these vulnerabilities.
SSH	CVE-2008-5161	Numerous vulnerability scanners report that the use of AES-CBC in SSH is not secure. See http://community.arubanetworks.com/t5/Unified-Wired-Wireless-Access/SSH-and-AES-CBC/m-p/248919 for a full explanation. Note that beginning in ArubaOS 6.5.4.4, it is possible to disable AES-CBC.
sssd	CVE-2009-2410	ArubaOS does not include this service
DNS / Nameserver	CVE-2008-1447	ArubaOS includes a “DNS responder” that listens on UDP port 53. Any query sent to this responder will result in a response that contains the controller’s IP address. Vulnerability scanners may report that this service responds to recursive queries, that it allows cache snooping, or that it enables traffic amplification attacks. It is important to note that this

		service is not an actual DNS server, and these warnings may be safely ignored.
Web server	CVE-2002-0840 CVE-2012-0053	Warnings such as the following may be reported: • "Apache server mod_info is publicly available" • Error page XSS using wildcard DNS • Warnings related to PHP • Warnings related to WebGlimpse • Warnings related to phpCMS • Warnings related to WordPress • Warnings related to ComicPress • Warnings related to SodaHead Very often, vulnerability scans are run against a mobility controller's Captive Portal interfaces, such as ports 8080, 8081, and 8088. These interfaces have unusual properties in that all requests result in a redirect/refresh. The request contents will be reflected back in the response, encapsulated into an HTTP "meta" refresh tag. A number of vulnerability scanners incorrectly flag these responses as Cross-Site Scripting (XSS) based on seeing script code in the request appear in the response (i.e. they do not parse the "meta" tag). Additionally, because *all* requests to these interfaces, regardless of URL, are answered (with a redirect/refresh), the vulnerability scanner incorrectly identifies a web application as being present on these ports. Some scanners report a vulnerability related to the HTTP CONNECT method being supported. ArubaOS is designed to work with web proxies for captive portal clients, and CONNECT methods will be redirected from port 8088 to the controller's own port 8080 or 8081. This is not a vulnerability, but rather expected behavior. Additionally, these scanners work by looking for a HTTP 2xx or 3xx response code in response to the CONNECT method. Because ArubaOS will issue a HTTP 200 response code in conjunction with "meta refresh" content, vulnerability scanners may be falsely triggered. Missing HTTP Strict-Transport-Security header has been corrected under bug 177420 and is fixed in version 6.5.4.13 and later. Missing HTTP Content-Security-Policy header is an intentional design choice (bug 177308). Use of this header caused interoperability problems with certain browsers, so it was removed. Aruba will revisit this header in the future to determine if compatibility problems have been resolved.
TLS / SSL	CVE-2009-3555 CVE-2011-3389	CVE-2011-3389 ("BEAST") is a client-side vulnerability and as such cannot be "fixed" on the server side. To mitigate, disable the use of TLS

	CVE-2013-0169 CVE-2014-0160 CVE-2014-0224 CVE-2014-3566 CVE-2014-8730 CVE-2015-4000 CVE-2016-2183	1.0 and TLS 1.1, as described in this guide. CBC ciphers may also be disabled on the client side as mitigation. CVE-2014-3566 ("Poodle") is a vulnerability in SSL 3.0. Older versions of ArubaOS allowed SSL 3.0 to be disabled; newer versions remove SSL 3.0 completely. CVE-2015-4000 ("Logjam") is not a vulnerability within ArubaOS. Numerous vulnerability scanners appear to report that all TLS servers are vulnerable, even when they are not. Some vulnerability scanners report that ArubaOS is vulnerable to CVE-2016-2183 ("SWEET32"). Since mid-2016, 3DES support was removed from ArubaOS, so this is a false positive.
TLS/SSL Certificate		Warnings such as the following may be reported: • Untrusted TLS/SSL server X.509 certificate • X.509 Certificate Subject CN Does Not Match the Entity Name • SHA-1 based Signature in TLS/SSL Server X.509 Certificate • TLS Server Certificate Modulus less than 2048 bits • SSL Certificate Name Mismatch ArubaOS ships with a factory-default X.509 certificate called "securelogin.arubanetworks.com". This certificate should not be used in production networks. Typically, warning messages produced by vulnerability scanners are related to this certificate. Administrator should properly install a unique X.509 certificate, as described in this guide, to eliminate these warnings. These warnings are NOT false positives, but action from the administrator is required to correct the problem.
IP stack	CVE-2004-0230 CVE-2002-0510 CVE-1999-0524	These items are labeled, respectively, as: • TCP sequence number approximation • UDP constant identification field / UDP IP ID Zero • TCP / ICMP timestamp response While by their nature these cannot be fixed, Aruba has judged that the risk of exploitation is low enough that no further action is required.
Cryptography		Warnings such as the following may be reported: • Weak MAC • Weak ciphers in use • TLS Server Supports DES and IDEA Cipher Suites • TLS Server Supports Cipher Block Chaining Ciphers • TLS Server Supports use of Static Key Ciphers • TLS Server Supports Insecure TLS 1.0 Please see the section below entitled "Cryptography" for more information. In general, Aruba has spent significant time researching

		various cryptographic protocols, and believes that products are using a secure configuration. In some cases, the *most* secure settings cannot be used, in order to achieve compatibility with a wide range of client systems (e.g. support for static key ciphersuites such as TLS_RSA_*) so the resulting configuration is a reasonable tradeoff. In other cases, the international security standard Common Criteria mandates which cipher suites must be supported.
jQuery	CVE-2020-11023 CVE-2020-11022 CVE-2020-7656 CVE-2019-11358 CVE-2019-5428 CVE-2017-16011 CVE-2014-6071 CVE-2012-6708 CVE-2011-4969	A number of Aruba UI components make use of the jQuery package. The versions of jQuery in use may contain cross-site scripting vulnerabilities. Aruba has analyzed these potential vulnerabilities and has found no attack vector that would allow the vulnerabilities to be triggered. ArubaOS 8.7 contains an upgraded version of the jQuery package; because of significant complexity involved in implementing the upgraded package, the change will not be backported to earlier ArubaOS versions.

Locking Down Services

Cryptography

ArubaOS employs cryptography as a part of several services, including Wi-Fi/WPA2, HTTPS, SSH, IPsec, and others. While an administrator has a great deal of flexibility in configuring IPsec services, other services tend to provide few or no options.

In the FIPS version of ArubaOS, all cryptographic services provided a minimum strength of 112 bits as mandated by FIPS 140-2. Services which provide less than 112 bits of security (such as RSA-1024, SHA1 for digital signatures, MD5, DES) may not be configured.

In non-FIPS versions of ArubaOS, there are no restrictions on minimum security strength. Algorithms such as DES (56-bits of strength) and MD5 (<64 bits of strength) are permitted to be used, although this is not the default configuration.

- Wi-Fi/WPA2: AES-CCMP specifies a key size of 128 bits. This is not configurable. Do not use TKIP for Wi-Fi encryption as it contains known security weaknesses.
- HTTPS: Ensure that "`web-server profile ciphers high`" is configured. This is the factory default setting (note: in ArubaOS-FIPS, this command is not present since 'high' is the only setting available.) With this setting enabled, only ciphersuites supporting 128-bit or higher symmetric keys are allowed.

  Beginning with ArubaOS 6.3.1, TLS 1.2 is supported and should be used whenever possible; this may require browser configuration to ensure that TLS 1.2 is enabled. For the strongest security configuration, TLS 1.0 and TLS 1.1 should be disabled. To enable *only* TLS 1.2 support, configure "`web-server profile ssl-protocol tlsv1.2`". Note that disabling TLS 1.0 and 1.1 may lead to compatibility problems with older browsers.

The following ciphersuites are supported. Note that ECDSA ciphersuites require installation of the Advanced Cryptography License and installation of an ECDSA server certificate.

TLS_RSA_WITH_AES_128_CBC_SHA

TLS_RSA_WITH_AES_256_CBC_SHA

TLS_DHE_RSA_WITH_AES_128_CBC_SHA

TLS_DHE_RSA_WITH_AES_256_CBC_SHA

TLS_RSA_WITH_AES_128_CBC_SHA256

TLS_RSA_WITH_AES_256_CBC_SHA256

TLS_DHE_RSA_WITH_AES_128_CBC_SHA256

TLS_DHE_RSA_WITH_AES_256_CBC_SHA256

TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256

TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384

TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256

TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384

- SSH: SSH ciphersuites are not configurable. Aruba recommends configuring SSH clients to support the highest strength desired. The following lists the default settings:
  - ArubaOS 6.x FIPS: aes128-cbc,aes256-cbc with hmac-sha1,hmac-sha1-96
  - ArubaOS 6.x (non-FIPS): aes128-ctr,aes192-ctr,aes256-ctr,arcfour256,arcfour128,aes128-cbc,3des-cbc,blowfish-cbc,cast128-cbc,aes192-cbc,aes256-cbc,arcfour,rijndael-cbc with hmac-md5,hmac-sha1,umac-64,hmac-ripemd160,hmac-ripemd160,hmac-sha1-96,hmac-md5-96
  - ArubaOS 8.x: aes128-ctr,aes192-ctr,aes256-ctr,aes128-cbc,aes256-cbc, hmac-sha1,hmac-sha1-96,hmac-sha2-256

  Some vulnerability scanners complain about the use of HMAC-SHA1-96 and AES-CBC, claiming that such use is weak. These vulnerability scanners are incorrect; it appears that the scanner vendors have an incomplete understanding of cryptography. A full explanation is outside the scope of this document, but truncating the output bits from HMAC does not produce a weaker MAC (see RFC 2104 and NIST SP800-107). In addition, note that weaknesses in SHA1 do not translate to HMAC-SHA1; the two are different algorithms with different properties (see NIST SP800-107 and SP800-131A).

  Beginning with ArubaOS 6.5.4.4, it is possible to restrict the use of specific SSH ciphers and MACs. By default, the available SSH ciphers are aes128-cbc,aes256-cbc,aes128-ctr,aes192-ctr,and aes256-ctr. By default, the available SSH MACs are hmac-sha1 and hmac-sha1-96. It is possible to disable any of these using the following configuration commands:

```
(Hostname) #configure terminal
Enter Configuration commands, one per line. End with CNTL/Z
(Hostname) (config) #ssh disable-ciphers aes-cbc
(Hostname) (config) #ssh disable-ciphers hmac-sha1-96
```

CPsec

Control Plane Security (CPsec) is a service that forces all control-channel traffic (such as configuration updates) between Aruba network elements to occur inside IPsec tunnels. The IPsec tunnels are authenticated using X.509 certificates provisioned onto each Aruba device during the manufacturing process. The private key for the certificate is stored inside a Trusted Platform Module (TPM), providing hardware protection and tamper-resistance. CPsec is typically deployed between APs and mobility controllers, although the term is also used to refer to IPsec tunnels between mobility controllers, such as those in a master-local cluster. Certificate-based authentication is much stronger than pre-shared key authentication, and because all Aruba network elements come pre-provisioned with certificates, CPsec is simple to enable.

IPsec

ArubaOS provides extensive configuration ability for IPsec. For use with CPsec and RAP termination, configuration is automatic. Configuration for other use cases will depend on a number of factors, including peer capabilities. In general, Aruba recommends use of IKEv2 and a security strength of at least 112 bits. The following configuration will provide approximately 112 bits of strength:

```
Protection suite priority 20
   Version 2
   encryption algorithm: AES - Advanced Encryption Standard (256 bit keys)
   hash algorithm: Secure Hash Algorithm 160
   authentication method: Rivest-Shamir-Adelman Signature
   PRF method: hmac-sha1
   Diffie-Hellman Group: #14 (2048 bit)
   lifetime: [300 - 86400] seconds, no volume limit

Crypto Map Template"default-ikev2-dynamicmap" 10000
         IKE Version: 2
         IKEv2 Policy: DEFAULT
         Security association lifetime seconds : [300 -86400]
         Security association lifetime kilobytes: N/A
         PFS (Y/N): Y (Use the 2048-bit Diffie Hellman prime modulus group
        Transform sets={ default-1st-ikev2-transform, default-3rd-ikev2-
        transform }

Transform set default-1st-ikev2-transform: { esp-aes256 esp-sha-hmac }
         will negotiate = { Transport, Tunnel }
Transform set default-3rd-ikev2-transform: { esp-aes128 esp-sha-hmac }
         will negotiate = { Transport, Tunnel }
```

The above configuration can be achieved by starting with the factory default configuration, and then issuing the following set of configuration commands:

```
(Hostname) #configure terminal

Enter Configuration commands, one per line. End with CNTL/Z

(Hostname) (config) #crypto isakmp policy 20
(Hostname) (config-isakmp)# version v2
(Hostname) (config-isakmp)# group 14
(Hostname) (config-isakmp)# authentication RSA-sig
(Hostname) (config-isakmp)# exit
```

```
(Hostname) (config) #crypto dynamic-map default-ikev2-dynamicmap 10000
(Hostname) (config-dynamic-map)# set pfs group14
```

Stronger configurations are available with installation of the Advanced Cryptography License. Through the use of Suite B cryptography, an overall strength of 128 and 192 bits is possible.

Diffie-Hellman

An integral part of IPsec and TLS is the Diffie-Hellman key exchange. In October, 2015, a group of researchers speculated that entities with nation-state resources may have the ability to break 1024-bit Diffie-Hellman parameters through a massive pre-computation attack. For details, see the paper at https://weakdh.org/imperfect-forward-secrecy-ccs15.pdf. Although this paper is speculative, Aruba has responded by beginning the process to move away from common 1024-bit DH groups.

- Beginning in ArubaOS 6.4-FIPS, use of the 1024-bit Diffie-Hellman (DH) group 2 for IPsec has been eliminated. All IKE/ISAKMP policies use DH group 14 (2048-bit). Two legacy IKE policies using DH group 2 are enabled by default in order to support upgrading from previous ArubaOS versions; these may be disabled once all components are running ArubaOS 6.4 or greater.
- Non-FIPS ArubaOS 6.x still uses DH group 2 by default, largely for performance and backwards compatibility reasons. However, beginning in ArubaOS 6.4, factory-default IKE policies (with priority numbers of 10000 or higher) may be disabled, and replaced with administrator-defined IKE policies.
- Beginning in ArubaOS 6.4.4.0, Access Points (APs) will attempt their connection using DH group 14 if their initial attempt using DH group 2 fails. To enable APs to use DH group 14, first disable the default AP ISAKMP policies. Then create a new policy, matching all parameters from the default except the DH group. When APs attempt to connect with DH group 2, the controller will respond with an INVALID_KE_PAYLOAD error message. This will cause the AP to retry the connection using DH group 14.
- ArubaOS 8.7 phases out use of DH group 2. This includes master-local and other CPsec communication. DH group 14 and HMAC-SHA256 are the new system defaults. See the ArubaOS 8.7 release notes for details of this change.
- Diffie-Hellman is also used during TLS key exchanges, when communicating with a mobility controller through HTTPS. In ArubaOS 6.4.4.0, the common DH group used by the Apache web server was replaced with a custom, Aruba-generated DH group. While a 1024-bit group is still being used, it is unique to Aruba and thus unlikely to be precomputed by a nation-state. As of ArubaOS 8.7, finite-field Diffie-Hellman (DHE) is no longer supported in TLS and only Elliptic Curve Diffie-Hellman (ECDHE) is supported.

Authenticated NTP

To ensure that audit logs contain accurate timestamps, it is critical that the system clock be correct. The Network Time Protocol (NTP) is typically used to synchronize clocks on Internet-connected devices. For a higher degree of security, NTP updates should be authenticated, to prevent an attacker from intercepting NTP communication and responding with false time data. To enable NTP authentication:

```
(Hostname) #configure terminal
(Hostname) (config) #ntp authentication-key 1 md5 VeryStrongPresharedKey
(Hostname) (config) #ntp trusted-key 1
(Hostname) (config) #ntp server 192.168.1.253 key 1
(Hostname) (config) #ntp authenticate

(Hostname) #show ntp status
```

```
Authentication:          enabled
```

SNMP

The Simple Network Management Protocol is commonly used by network management systems to poll devices for information such as port configuration, status, and interface counters. But SNMP versions 1 and 2 provide very little security beyond the community string. If an attacker has network access to a device and can guess the community string, it may lead to disclosure of sensitive information. Aruba strongly recommends the use of SNMPv3, which includes much stronger security through authentication and encryption. To configure SNMPv3, add an SNMP user as shown below. (Note: For a full explanation of SNMP configuration, please consult the ArubaOS User Guide.)

```
(Hostname) #configure terminal
(Hostname) (config) #snmp-server user "snmpuser" auth-prot SHA "ReallyStrongAuthPassword"
priv-prot AES "ReallyStrongEncryptionPassword"
(Hostname) (config) #snmp-server host "10.1.1.1" version 3 "snmpuser"
```

External syslog

In the event that a system is compromised, one of the first things an attacker will often do is to remove evidence of the intrusion from the system logs. For this reason, it is important to send logs to an external system – preferably one with automated log analysis tools that can identify and flag unusual activity. ArubaOS supports industry-standard “syslog” for this purpose. External logging configuration supports a number of filters and options, but enabling the sending of all logs to an external server is straightforward:

```
(Hostname) #configure terminal
Enter Configuration commands, one per line. End with CNTL/Z

(Hostname) (config) #logging 10.1.1.2
```

RADIUS secrets

The RADIUS protocol provides a weak form of encryption, which uses the RADIUS shared secret as the basis for the encryption key. Ensure that the RADIUS shared secret is as long and as complex as possible – ArubaOS supports a maximum length of 63 characters. There is no need for this secret to be memorable by a human, so use a service such as http://www.random.org/ to generate a truly random string.

An authentication server performing authentication for WPA2 sessions will use the RADIUS protocol to send the WPA2 Pairwise Master Key (PMK) to an Aruba mobility controller – an attacker intercepting this key would also be able to monitor and decrypt Wi-Fi traffic over the air. If the link between the RADIUS server and the Aruba device is trusted (e.g. within the same datacenter) then relying on RADIUS encryption is sufficient. However, if the path traverses untrusted segments, such as WAN links, RADIUS traffic should be secured inside IPsec tunnels.

Banner Messages

Banners are electronic messages that provide notice of legal rights to users of computer networks. Banners may be used to generate consent to real-time monitoring, and to remove legal protection against an expectations of privacy that a user might have when accessing a system. The exact text to

be used in a banner message should be provided by a competent attorney, but some samples are available from http://www.cio.ca.gov/OIS/Government/library/documents/BannerSamples.doc.

In ArubaOS, a banner message will be displayed prior to an SSH login attempt, and also prior to a WebUI login attempt. To configure a banner message:

```
(Hostname) (config) #banner motd $
Enter TEXT message [maximum of 4095 characters].
Each line in the banner message should not exceed 255 characters.
End with the character '$'.

This is a Department of Defense (DoD) computer system.  DoD computer systems
are provided for the processing of Official U.S. Government information only.
All data contained within DoD computer systems is owned by the Department of
Defense, and may be monitored, intercepted, recorded, read, copied, or
captured in any manner and disclosed in any manner, by authorized personnel.

THERE IS NO RIGHT OF PRIVACY IN THIS SYSTEM.  System personnel may disclose
any potential evidence of crime found on DoD computer systems for any reason.
USE OF THIS SYSTEM BY ANY USER, AUTHORIZED OR UNAUTHORIZED, CONSTITUTES
CONSENT TO THIS MONITORING, INTERCEPTION, RECORDING, READING, COPYING, or
CAPTURING and DISCLOSURE.
$
```

Wireless IDS

Aruba mobility controllers include basic rogue AP detection in all systems, and much more advanced capabilities when the RFProtect license has been purchased. A full description of WIDS/WIPS configuration is beyond the scope of this document. However, the ArubaOS WebUI provides a “WIP Wizard” which eases configuration of WIDS features significantly. To run this wizard, access the WebUI “Configuration” page, then navigate to Wizards->WIP. Setting levels to “Medium” provides a good baseline level of protection, without creating too many opportunities for false positives. Setting levels to “High” will enable all features, but may increase false positives until the system is tuned for the local environment.

Control Path Defense

A number of features are available to defend the ArubaOS control path from attack. The current settings can be viewed by using the "`show firewall`" command. Relevant output includes:

```
Monitor/police CP attacks                    Enabled   20/30sec
Rate limit CP untrusted ucast traffic        Enabled   20 Mbps
Rate limit CP untrusted mcast traffic        Enabled   4 Mbps
Rate limit CP trusted ucast traffic          Enabled   160 Mbps
Rate limit CP trusted mcast traffic          Enabled   4 Mbps
Rate limit CP route traffic                  Enabled   2 Mbps
Rate limit CP session mirror traffic         Enabled   2 Mbps
Rate limit CP auth process traffic           Enabled   2 Mbps
```

Traffic destined for an IP address belonging to the mobility controller itself will be subject to these rules. For example, unicast traffic originating from an untrusted interface and destined for the controller's management interface will be rate-limited to 20Mbps using the default configuration above. To configure the "Monitor/police CP attacks" value, the following command was used:

```
(Hostname) (config) #firewall attack-rate cp 20
```

Consult the ArubaOS User Guide for more detailed information on these settings.

Locking Down Administrative Access

A primary attack vector for intrusions is often the administrative console – in ArubaOS, the graphical WebUI or the command-line. Both should be properly secured in order to ensure attackers are unsuccessful.

Console Port and Password Recovery

The serial console port on an Aruba Mobility Controller is considered a privileged interface. An attacker with physical access to the console port can easily compromise the controller using the password recovery procedure, which is widely known and is given out by Aruba Technical Support frequently when customers lose access to administrative passwords. The password recovery procedure can be used to change the password for the "admin" account – it does not reboot the controller or change any other configuration.

If good physical security cannot be provided for an Aruba controller, Aruba recommends applying tamper-evident labels (TELs) across the console port. TELs can be obtained from a wide variety of commercial sources. TELs are only effective if the controller is routinely inspected, however, to see if the label has been removed.

Beginning with ArubaOS 8.0, it is possible to disable the console port through software. This will prevent the password recovery procedure from being available. However, this command only disables the serial port while ArubaOS is running. If the controller is rebooted, the boot ROM may still be accessed through the console port, which can be used to boot alternative configuration files, boot alternative software images, or to erase data from flash memory. To disable console port access from ArubaOS:

```
(Hostname) #configure terminal
Enter Configuration commands, one per line. End with CNTL/Z


(Hostname) (config) # mgmt-user console-block
```

AP Console Port and Debug Access

In older versions of ArubaOS, the RS-232 console port on access points allowed access to a limited command shell without authentication. Modern versions automatically enable AP console protection by setting "`ap system-profile ap-console-protection`", and a console password, which is defined in "`ap system-profile ap-console-password`". The default setting is a randomly generated value that may be changed by the administrator. The console port may also be disabled completely by setting "`no console-enable`" in the AP system profile.

Remote access to an AP command shell for debugging may be enabled. This feature should be disabled when not in use. To enable the debugging shell, configure ap system-profile telnet. In older versions of ArubaOS, this would cause each AP to enable a telnet server and listen for connections on port 23. Starting in ArubaOS 8.7, telnet is replaced with SSH, even though the configuration setting has not changed from "telnet".

Access Control

Aruba recommends using firewall rules to permit administrative access only from authorized sources. On a wired network, this could be a particular subnet which is only used by authorized administrators.

For wireless users, make use of Aruba's role-based access control to place administrators into a unique role which has firewall rules that permit administrative access to the mobility controller. If network design permits, it may also be wise to create a dedicated management network that carries only network management traffic; only the management interface on the controller would permit administrative access.

The easiest way to implement access control for administrative access is use of the "service ACL". This is a set of firewall policies that affect all network traffic destined to any control plane element within the controller. While service ACLs can be used to control any network traffic destined for the controller, it is mostly commonly used for administrative access. By default, all traffic for ArubaOS control interfaces is permitted from any destination – these rules can be viewed by executing the "`show firewall-cp internal`" command:

```
(Hostname) #show firewall-cp internal

CP firewall policies
--------------------
IP Version  Source IP  Source Mask  Protocol  Start Port  End Port  Permit/Deny  hits  contract
----------  ---------  -----------  --------  ----------  --------  -----------  ----  --------
ipv4        any                     6         1723        1723      Permit       0
ipv4        any                     17        1701        1701      Permit       0
ipv4        any                     6         23          23        Deny         0
ipv4        any                     6         8084        8084      Deny         0
ipv4        any                     6         3306        3306      Deny         0
ipv4        any                     17        8209        8209      Deny         0
ipv4        any                     6         8211        8211      Deny         0
ipv4        any                     6         2300        2300      Permit       0
ipv4        any                     6         2323        2323      Permit       0
ipv4        any                     6         8211        8211      Permit       0
ipv4        any                     6         21          21        Permit       0
ipv4        any                     6         22          22        Permit       78
ipv4        any                     6         17          17        Permit       0
ipv4        any                     17        514         514       Permit       0
ipv4        any                     50        0           65535     Permit       0
ipv4        any                     17        8200        8200      Permit       0
ipv4        any                     112       0           65535     Permit       0
ipv4        any                     89        0           65535     Permit       0
…
```

These default rules will be *overridden* by any administrator-configured rules. So, for example, while the output above shows TCP traffic destined for port 22 (SSH) to be permitted from any source IP, an administrator may lock this traffic down further by specifying additional rules using the "`firewall cp`" configuration commands. The following example demonstrates restricting of SSH traffic only to a specific subnet:

```
(Hostname) # configure terminal
Enter Configuration commands, one per line. End with CNTL/Z

(Hostname) (config) #firewall cp
(Hostname) (config-fw-cp) #ipv4 permit 10.2.14.0 255.255.255.0 proto ssh
```

```
(Hostname) (config-fw-cp) #ipv4 deny any proto ssh
(Hostname) (config-fw-cp) #show firewall-cp

CP firewall policies
--------------------
IP Version  Source IP  Source Mask    Protocol  Start Port  End Port  Permit/Deny  hits  contract
----------  ---------  -----------    --------  ----------  --------  -----------  ----  --------
ipv4        10.2.14.0  255.255.255.0  6         22          22        Permit       0
ipv4        any                       6         22          22        Deny         0
```

The next example shows restricting access to TCP port 4343 (used by the Web-based administrative interface) only to a specific host:

```
(Hostname) # configure terminal
Enter Configuration commands, one per line. End with CNTL/Z

(Hostname) (config) #firewall cp
(Hostname) (config-fw-cp) #ipv4 permit host 10.2.14.5 proto 6 ports 4343 4343
(Hostname) (config-fw-cp) #ipv4 deny any proto 6 ports 4343 4343
(Hostname) (config-fw-cp) #show firewall-cp

CP firewall policies
--------------------
IP Version  Source IP  Source Mask      Protocol  Start Port  End Port  Permit/Deny  hits  contract
----------  ---------  -----------      --------  ----------  --------  -----------  ----  --------
ipv4        10.2.14.5  255.255.255.255  6         4343        4343      Permit       134
ipv4        any                         6         4343        4343      Deny         0
```

The service ACL definition also includes pre-defined services for HTTP, HTTPS, FTP, telnet, and others.

Password Policy

Authentication with username/password does not provide the strongest form of security, yet it is extremely common. When dealing with passwords, ArubaOS provides a number of tools that help ensure administrative passwords are managed correctly. These tools are collected under the "password-policy" configuration section, and may be viewed using the "`show aaa password-policy mgmt`" command. By default, password-policy is disabled, meaning any administrative password is accepted. Aruba recommends the following as a reasonably strong password policy:

```
(Hostname) #configure terminal
Enter Configuration commands, one per line. End with CNTL/Z

(Hostname) (config) #aaa password-policy mgmt.
(Hostname) (Mgmt Password Policy) #enable
(Hostname) (Mgmt Password Policy) #password-min-length 8
(Hostname) (Mgmt Password Policy) #password-not-username
(Hostname) (Mgmt Password Policy) #password-lock-out 3
(Hostname) (Mgmt Password Policy) #password-min-special-character 1
(Hostname) (Mgmt Password Policy) #password-min-digit 1
(Hostname) (Mgmt Password Policy) #password-min-uppercase-characters 1
(Hostname) (Mgmt Password Policy) #show aaa password-policy mgmt
```

```
Mgmt Password Policy
--------------------
Parameter                                                          Value
---------                                                          -----
Enable password policy                                             Yes
Minimum password length required                                   10 characters
Minimum number of Upper Case characters                            1 characters
Minimum number of Lower Case characters                            0 characters
Minimum number of Digits                                           1 digits
Minimum number of Special characters (!, @, #, $, %, ^, &, *,...)  1 characters
Username or Reverse of username NOT in Password                    Yes
Maximum consecutive character repeats                              0 characters
Maximum Number of failed attempts in 3 minute window to lockout user  3 attempts
Time duration to lockout the user upon crossing the "lock-out" threshold  3 minutes
```

Centralized Authentication and Authorization

In an organization where multiple administrators exist, who potentially have multiple privilege levels, the use of centralized authentication helps to prevent insider attacks. With centralized authentication, Aruba controllers do not have local administrative accounts. Instead, administrative users login with credentials that are authenticated remotely by a RADIUS or TACACS+ server. This authentication process can also potentially return role information back to the Aruba device, allowing the user to be placed into an administrative privilege level (e.g. 'root' or 'read-only').

Centralized authentication is enabled through "`aaa authentication mgmt`" configuration commands. The following configuration excerpt shows a common setup for authenticating administrative users against a RADIUS server.

```
(Hostname) #configure terminal
Enter Configuration commands, one per line. End with CNTL/Z
(Hostname) (config) #aaa authentication-server radius ClearPass
(Hostname) (RADIUS Server "ClearPass") #host 10.2.72.20
(Hostname) (RADIUS Server "ClearPass") #key ReallyStrongRADIUSSecret
(Hostname) (RADIUS Server "ClearPass") #exit
(Hostname) (config) #aaa server-group ClearPass_Group
(Hostname) (Server Group "ClearPass_Group") #auth-server ClearPass
(Hostname) (RADIUS Server "ClearPass") #exit
(Hostname) (config) #aaa authentication mgmt
(Hostname) (Management Authentication Profile) #enable
(Hostname) (Management Authentication Profile) #mschapv2
(Hostname) (Management Authentication Profile) #server-group ClearPass_Group
```

Once the configuration above has been entered, the local "admin" account should be set with a randomly-generated password to prevent anyone from using it. In an emergency, the "admin" password may be reset by accessing the serial console port and executing the lost password procedure, but under normal circumstances nobody will know the password to the local admin account.

Alternatively, local authentication may be completely disabled through the following command:

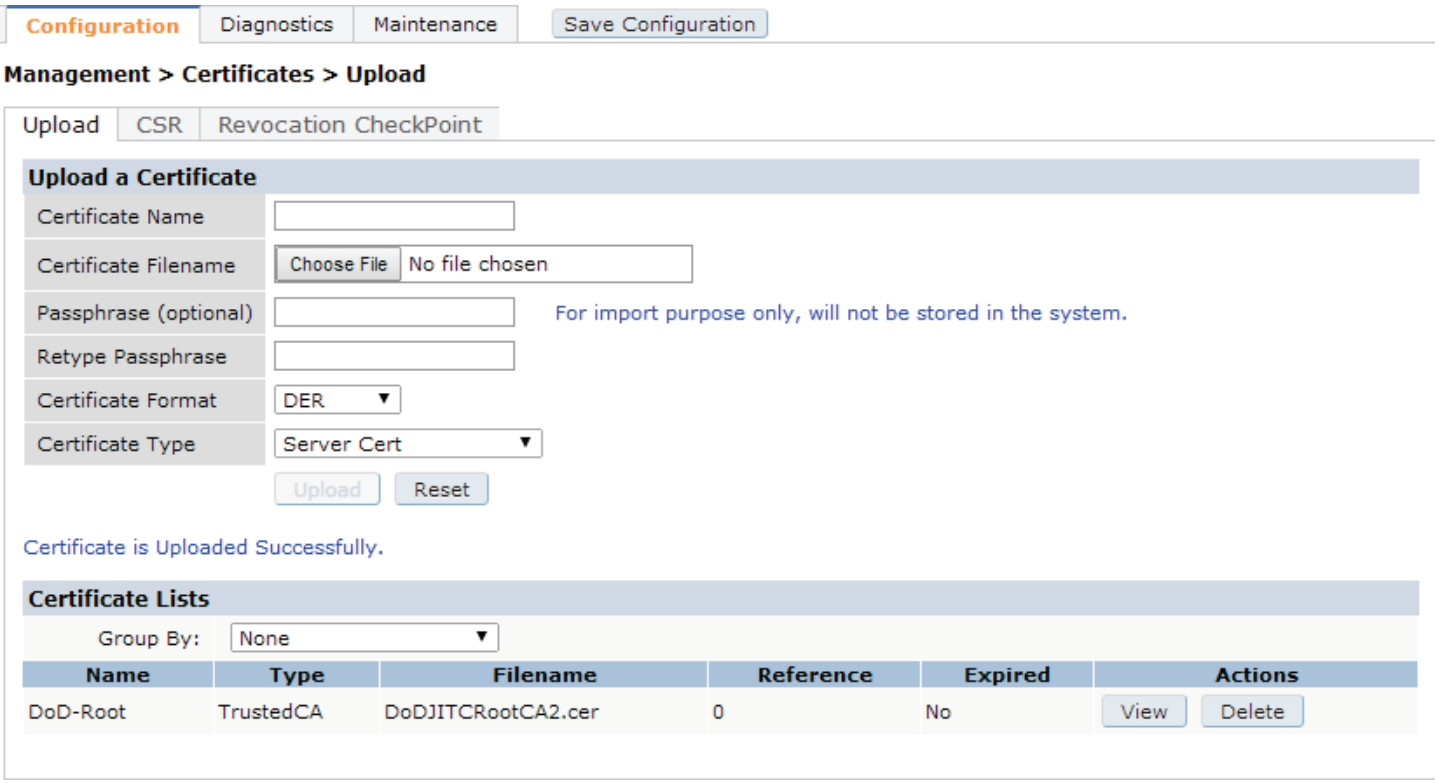
```
(Hostname) (config) #mgmt-user localauth-disable
```

Note that if local authentication is disabled, nobody will be able to access the controller's management interface in the event the authentication server becomes unavailable. For this reason, redundant authentication servers should always be used if local authentication is disabled. Because of the difficulty in regaining access should the authentication server become unavailable, Aruba prefers the "randomly created admin password" method to disable local authentication.

Strong Administrative Authentication

Because password-based authentication is generally considered to be weak, Aruba recommends the use of PKI-based authentication for high-security installations. PKI-based authentication may also incorporate two-factor authentication. Although Aruba devices recognize and support certificate-based authentication, they do not directly enforce a requirement for two-factor authentication – this happens on the client side. This can be achieved by requiring a PIN or password to get access to a certificate's private key. The most commonly-used form of two-factor authentication with Aruba devices is the smartcard, with the smartcard protected by entry of a PIN to unlock access to a private key.

Certificate-based authentication is available for both HTTPS/WebUI and SSH/CLI. To get started, a trusted root CA certificate must be uploaded through the WebUI (Configuration->Certificates). This is the CA to which all administrator end-user certificates must chain.



Once the CA certificate has been configured, configure the system to use certificate-based authentication for WebUI access. There are two methods which may be used. To enable access for a **specific certificate**, add a management user and specify the serial number of the certificate:

```
(Hostname) (config) #mgmt-user webui-cacert "DoD-Root" serial "12345" "jon" "root"
(Hostname) (config) #web-server
(Hostname) (Web Server Configuration) #mgmt-auth certificate
```

The command above adds a username “jon” with a role of “root” who will be authenticated using a certificate issued by the CA (previously loaded) “DoD-Root” and having a serial number of “12345”. Each additional user who is authorized to login to the WebUI should be added in a similar manner.

Note that once “`mgmt-auth certificate`” is configured, username/password authentication will no longer be supported for WebUI logins. If both certificate-based and username/password-based authentication needs to be supported, use the configuration statement “`mgmt-auth certificate username/password`”.

The second method enables access for any user who has a valid certificate and is authorized by an external authentication server (RADIUS or LDAP). To configure this form of access:

```
(Hostname) (config) #mgmt-user webui-cacert DoD-Root
(Hostname) (config) #web-server
(Hostname) (Web Server Configuration) #mgmt-auth certificate
```

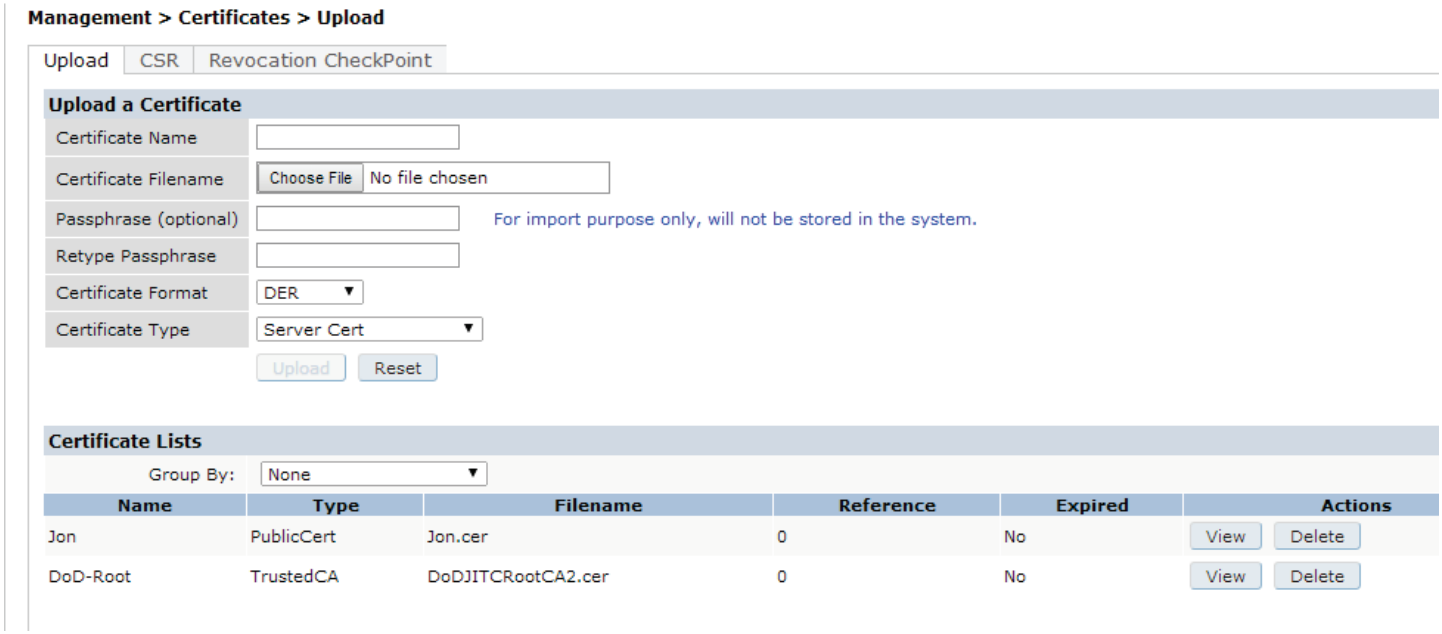
In addition to this configuration statement, an external authentication server must be configured as specific above in the “Centralized Authentication and Authorization” section. With this configuration, the authentication sequence is as follows:

1. User connects browser to https://controller.ip.address
2. Controller begins TLS handshake and requests certificate from client
3. Client computer requests user to select a certificate (optionally requests PIN/passcode to access private key)
4. Client responds with certificate
5. Controller validates that the user’s certificate chains up to a trusted root CA
    a. Optional: If the controller has been configured to perform revocation checking with OCSP, the user’s certificate will be checked for revocation status against an OCSP responder.
6. Controller will extract the Subject Alternative Name:Principal Name field from the certificate to be used for an authorization check.
    a. If RADIUS is used as the authentication server, the controller will generate a RADIUS “authorize-only” transaction using the Principal Name. Note that an authorize-only transaction is different than a standard RADIUS authentication request, and the RADIUS server must be configured to support this type of request. Not all RADIUS servers support authorize-only; ClearPass Policy Manager is one RADIUS server that *does* support authorize-only.
    b. If LDAP is used as the authentication server, the controller will issue an LDAP request using the Principal Name as the key attribute. Note that the LDAP server configuration in ArubaOS should contain ‘`key-attribute "userPrincipalName"`’ for this to work.
7. If the authentication server responds with a “success” message, the user is authenticated and given the default role as defined in the “`aaa authentication mgmt`" profile (unless role information is returned by the authentication server.)

ArubaOS also supports authentication of SSH sessions using certificates, but the process is slightly more complex. Until recently, there was no standard for X.509-based authentication of SSH sessions. At the time of this writing, RFC 6187 has been standardized, but no SSH clients have yet been

developed which support this standard. ArubaOS supports an older method of X.509 authentication known as "x509v3-sign-rsa" specified in the IETF draft draft-ietf-secsh-x509-03. Support for this specification is available in OpenSSH (with a patch applied), SecureCRT, and Tectia SSH. At a future date, Aruba plans to add support for RFC 6187, as long as interoperability can be assured with common SSH clients.

To configure a system for certificate-based SSH authentication, load a trusted CA certificate as above. A revocation checkpoint should also be configured to ensure that a certificate has not been revoked. Once a trusted CA has been loaded, each authorized user's certificate must be uploaded to the controller, uploaded as a "PublicCert":



Note that the uploaded user's certificate should NOT contain the private key – only the public portion of the certificate should be uploaded. Once certificates have been loaded, the following configuration may be used to enable SSH access:

```
(Hostname) #configure terminal
Enter Configuration commands, one per line. End with CNTL/Z

(Hostname) (config) #ssh mgmt-auth public-key
(Hostname) (config) #mgmt-user ssh-pubkey client-cert "Jon" "jon" "root" rcp "DoD-Root"
```

The above configuration adds a user named "jon" with a role of "root". This user must have a private key corresponding to the uploaded certificate named "Jon", and the controller will check for revocation status using the revocation checkpoint configured for "DoD-Root". Note that by configuring "`ssh mgmt-auth public-key`", username/password authentication will no longer be accepted.

Locking Down User Access

User Roles and Firewall Policies

Aruba recommends deployment of role-based access control for wireless users. Rather than granting one-size-fits-all access to the network once users have authenticated, only grant access appropriate for that user's role in the organization. An example previously given was preventing non-administrative users from gaining access to the administrative access interfaces on network equipment. Another example includes preventing wireless users from running local network services, such as DHCP servers or web servers. There is no single approach that works for all organizations; administrators will need to evaluate their own needs and requirements.

There is extensive documentation available covering the configuration of roles and firewall policies. See the ArubaOS User Guide at http://support.arubanetworks.com or the Campus Validated Reference Design at http://www.arubanetworks.com/vrd/CampusWNetworksVRD for examples and configuration guidance.

Note that many factory-default firewall rules may NOT be appropriate for a secure network. For example, the "logon-control" ACL permits NAT-T (IPsec over UDP port 4500), and the default captive portal role includes the "logon-control" ACL. This would have the effect of letting captive portal users establish IPsec sessions without logging in. **Examine default rules carefully** by using the "show rights" command, and edit out those that are not necessary.

Valid IP Address ACL

ArubaOS defines a default firewall policy called "validuser", which prevents invalid IP addresses from appearing in the user table (either accidentally or maliciously). From a security standpoint, insertion of a rogue IP address into the user table could cause a denial of service attack – for example, if a wireless user were to statically define his IP address to be the same address as the local DNS server. The default validuser ACL is as follows:

```
ip access-list session validuser
  network 127.0.0.0 255.0.0.0 any any  deny
  network 169.254.0.0 255.255.0.0 any any  deny
  network 224.0.0.0 240.0.0.0 any any  deny
  host 255.255.255.255 any any  deny
  network 240.0.0.0 240.0.0.0 any any  deny
  any any any  permit
  ipv6 host fe80:: any any  deny
  ipv6 network fc00::/7 any any  permit
  ipv6 network fe80::/64 any any  permit
  ipv6  alias ipv6-reserved-range any any  deny
  ipv6  any any any  permit
```

Using the default settings, certain IP address ranges are prevented from appearing in the user table – for example, the IP multicast address range of 224.0.0.0. Aside from the specific "deny" entries, all other IP addresses are permitted for wireless users. For maximum security, the validuser ACL should be modified so that ONLY IP addresses valid for wireless clients are permitted. One way to do this is to manually specify entries that should be permitted. For example, if all wireless clients belong in the subnet 192.168.15.0/24, the following modification to the default ACL should be made:

```
(Hostname) (config) #ip access-list session validuser
```

```
(Hostname) (config-sess-validuser)#network 192.168.15.0 255.255.255.0 any any permit
(Hostname) (config-sess-validuser)#no any any any permit
(Hostname) (config-sess-validuser)#any any any deny
(Hostname) (config-sess-validuser)#show ip access-list validuser
```

Verify the output using “show ip access-list validuser”, noting that IPv4 entries are indicated by “4” and IPv6 entries are indicated by “6”. Although IPv4 and IPv6 rules appear in the same ACL, they are processed separately.

The “validuser” ACL should not be applied to an interface or role. This ACL is automatically referenced each time a user is added to the user-table.

Another, simpler, method to configure the same functionality is through the use of the “local-valid-users” feature in the global firewall configuration. If this feature is enabled, *only* IP addresses within the locally-defined VLAN interfaces address space will be permitted in the user table. As long as the controller has an IP interface defined in every wireless subnet (i.e. an “`interface vlan x; ip address a.b.c.d`” configuration statement for each wireless VLAN), this feature works automatically. If the controller does not have an IP interface in each wireless subnet, use the manual ACL method described above. To enable “local-valid-users”:

```
(Hostname) (config) #firewall local-valid-users
```

Global Firewall Settings

A number of firewall options are available which apply to all user traffic on the system. This section discusses a few of these settings which can be important when locking down security settings. Other settings are available (consult the ArubaOS User Guide for details) but these are considered to be the most useful/valuable for locking down security controls.

Preventing Inter-User Traffic

When this setting is enabled, wireless users are prevented from communicating with each other. All traffic originating from a wireless user, destined for another wireless user, is dropped. Note that this option may have significant impacts on network behavior; all forms of peer-to-peer communication are interrupted. To enable the feature:

```
(Hostname) (config) #firewall deny-inter-user-traffic
```

Note that the same feature is available on a per-SSID basis (technically, a per-virtual-AP basis) by configuring the setting in a virtual-ap profile:

```
(Hostname) (config) #wlan virtual-ap default
(Hostname) (Virtual AP profile "default") #deny-inter-user-traffic
```

A related feature will block only non-IP traffic, but will permit IP traffic between users (subject to firewall policies that have been applied to the user role.) This is a less-restrictive option than the previous setting. Because ARP traffic is considered non-IP, this setting will also disrupt ARP between wireless clients. For this reason, you may wish to enable proxy ARP on the user VLANs, which will cause the controller to proxy-ARP on behalf of wireless users.

```
(Hostname) (config) #firewall deny-inter-user-bridging
(Hostname) (config) #interface vlan 1
(Hostname) (config-subif)#ip local-proxy-arp
```

The use of "`interface vlan 1; bcmc-optimization`" will also enable proxy-arp; consult the ArubaOS User Guide to determine which feature is more appropriate to use.

Prohibit IP Spoofing

An IP spoofing attack occurs when a wireless user statically configures a device with an IP address belonging to another device on the network. ArubaOS includes a feature that locks a particular IP address with a particular MAC address. Enabling this feature accomplishes two main goals:

1. A man-in-the-middle attack using the so-called "Hole 196" vulnerability is defeated. In "Hole 196", an authenticated WPA2 user makes use of the Group Transient Key (GTK) to send unicast data to another wireless client. If this traffic includes gratuitous ARP packets for the default gateway's IP address, the attacker may be able to fool another wireless client into directing all off-net IP traffic to the attacker's workstation. When prohibit-ip-spoofing is enabled, the ARP poisoning attack through the "Hole 196" vulnerability could still succeed. However, the controller would then recognize traffic being sent to the attacker's MAC address with a destination IP address that does not match the attacker's assigned IP address. This traffic would then be dropped.
2. An attacker would be prohibited from cloning another user's IP address.

To enable the prohibit-ip-spoofing feature:

```
(Hostname) (config) #firewall prohibit-ip-spoofing
```

Prohibit ARP Spoofing

The prohibit-arp-spoofing feature blocks a wireless client from sending ARP responses, or gratuitous ARP messages, for an IP/MAC combination that is not its own. To enable:

```
(Hostname) (config) #firewall prohibit-arp-spoofing
```

Enforce DHCP

In an environment where DHCP is used for wireless client IP address assignment, a feature is available which forces wireless clients to use DHCP; static IP address assignments would not be supported. When this feature is enabled, traffic is dropped from clients that do not complete a DHCP negotiation, and from clients who are assigned an IP address through DHCP but then attempt to use a different IP address. The feature is enabled through the AAA profile – if there are multiple AAA profiles in use on the controller, it must be enabled for each one.

```
(Hostname) (config) #aaa profile default
(Hostname) (AAA Profile "default") #enforce-dhcp
```

User Authentication

Authentication is a critical component of wireless security. Aruba unequivocally recommends the use of **WPA2 with EAP-TLS** authentication to provide the strongest security posture. EAP-TLS is certificate-based, where both sides of the connection validate that the other side possesses the private key for a trusted certificate. The method is not subject to man-in-the-middle attacks, and does not depend on an encrypted TLS tunnel to provide security for a weak inner authentication exchange.

- Aruba controllers support EAP Termination, whereby the controller itself processes all 802.1X authentication and performs credential validation. For EAP-TLS, this includes validating that a client certificate is trusted and has not been revoked. However, EAP Termination was developed at a time when RADIUS servers in common use had limited support for 802.1X. Today, that situation has changed, and Aruba recommends use of an external authentication server such as ClearPass Policy Manager for 802.1X authentication. A purpose-built authentication server provides much more flexibility in authentication and authorization policies.

Appendix A: Open-Source Software Manifest

ArubaOS makes use of a number of open-source software packages. The list below, current as of ArubaOS 6.5, lists the types and version numbers of security-relevant (i.e. exposed to attack) open-source packages that are embedded in ArubaOS. Note that as explained previously, these open-source packages may have been patched against specific security vulnerabilities already, even though the version number was not incremented. In addition, not every feature of every package is enabled in ArubaOS. Therefore, a reported vulnerability in one of the following packages does not necessarily translate directly into an ArubaOS vulnerability.

Name	Version
Apache httpd	2.4.3
curl	7.26
dnsmasq	2.55
dhcpd (ISC)	4.1-ESV-R8
l2tpd	0.67
Linux Kernel	2.6.35
ncftp	3.1.5
Ntpd	4.2.6p5a
OpenLDAP	2.4.24
OpenSSH	5.8-p1
OpenSSL	1.0.1c
pppd	2.4.1
pptpd	1.1.3
SSHTerm	0.2.2
TACACS+	1.6.9
U-boot	1.1.4
wpa_supplicant	0.5.7, 0.6.10
wget	1.10.2
xyssl	0.9

For More Information

The best source of information on Aruba products, outside of official documentation, is the Airheads Social community. For security-related discussions, please visit the “AAA, NAC, and Guest Access” forum at http://community.arubanetworks.com/.

aruba
a Hewlett Packard Enterprise company