The ATM Forum

Technical Committee

Methods for Securely Managing ATM Network Elements - Implementation Agreement

Version 1.0

AF-SEC-0179.000

April 2002

The information in this publication is believed to be accurate as of its publication date. Such information is subject to change without notice and The ATM Forum is not responsible for any errors. The ATM Forum does not assume any responsibility to update or correct any information in this publication. Notwithstanding anything to the contrary, neither The ATM Forum nor the publisher make any representation or warranty, expressed or implied, concerning the completeness, accuracy, or applicability of any information contained in this publication. No liability of any kind shall be assumed by The ATM Forum or the publisher as a result of reliance upon any information contained in this publication.

The receipt or any use of this document or its contents does not in any way create by implication or otherwise:

- Any express or implied license or right to or under any ATM Forum member company's patent, copyright, trademark or trade secret rights which are or may be associated with the ideas, techniques, concepts or expressions contained herein; nor
- Any warranty or representation that any ATM Forum member companies will announce any product(s) and/or service(s) related thereto, or if such announcements are made, that such announced product(s) and/or service(s) embody any or all of the ideas, technologies, or concepts contained herein; nor
- Any form of relationship between any ATM Forum member companies and the recipient or user of this document.

Implementation or use of specific ATM standards or recommendations and ATM Forum specifications will be voluntary, and no company shall agree or be obliged to implement them by virtue of participation in The ATM Forum.

The ATM Forum is a non-profit international organization accelerating industry cooperation on ATM technology. The ATM Forum does not, expressly or otherwise, endorse or promote any specific products or services.

NOTE: The user's attention is called to the possibility that implementation of the ATM interoperability specification contained herein may require use of an invention covered by patent rights held by ATM Forum Member companies or others. By publication of this ATM interoperability specification, no position is taken by The ATM Forum with respect to validity of any patent claims or of any patent rights related thereto or the ability to obtain the license to use such rights. ATM Forum Member companies agree to grant licenses under the relevant patents they own on reasonable and nondiscriminatory terms and conditions to applicants desiring to obtain such a license. For additional information contact:

The ATM Forum
Worldwide Headquarters
572 B Ruger Street
San Francisco, CA 94129-0920
Tel: +1.415.561.6110
Fax: +1.415.561.6120

Acknowledgments

The production of this specification would not be possible without the enormous amount of effort provided by many individuals. Special acknowledgements for their hard work and dedication go to Richard Graveman and Wolfgang Klasen, the current chair and vice-chair.

In addition, the following individuals (listed alphabetically), among others, contributed their time and expertise to the development of this specification:

Kim Hebda
Chris Kubic
Michael Pierce

Gary Buda
Editor, ATM Forum Security Working Group

TABLE OF CONTENTS

1.0 Introduction

This Implementation Agreement lists, profiles, and describes a set of existing security standards and their use for securing access to ATM Network Elements for management, administration, operations, maintenance, and related tasks. Its goals are to define:

?? Uniform security services and mechanisms in a multi-vendor network,

?? Consistent key management, and

?? Controlled, consistent identification, authentication, and authorization of network administrators.

This specification focuses on how to apply popular security mechanisms and procedures (e.g., Kerberos and SSL) to the problem of securely managing an ATM Network Element. It does not develop any new protocols, and it does not model security features with new MIB objects.

If an ATM Network Element provides any of the management access methods listed below, then it shall support the corresponding security services and mechanisms described in this Implementation Agreement:

?? Command line interface via serial communications or telnet,

?? Web access via HTTP,

?? SNMP access to a SNMP agent.

2.0 General Security Objectives

A vendor that addresses security for its management, administration, operations, and maintenance interfaces between its ATM Network Elements and a Network Management System should consider the following list of security objectives and state which are met by its products:

2.1 Confidentiality

C-1 The interface between the ATM Network Element and the Network Management System shall support confidentiality of communications between the ATM Network Element and the Network Management System.

C-2 The interface between the ATM Network Element and the Network Management System shall support the confidentiality of passwords and key material.

C-3 The ATM Network Element shall support confidentiality of audit information.

C-4 The ATM Network Element shall support confidentiality of configuration files.

C-5 The ATM Network Element shall provide confidentiality for the storage of active and inactive passwords and key material.

C-6 The ATM Network Element shall provide confidentiality of addressing information.

2.2 Data Integrity

I-1 The interface between the ATM Network Element and the Network Management System shall support integrity for communications between the ATM Network Element and the Network Management System.

I-2 The interface between the ATM Network Element and the Network Management System shall support a mechanism for replay protection for communications with the Network Management System.

I-3 The ATM Network Element shall support integrity of configuration files.

I-4 The ATM Network Element shall support integrity of audit information.

2.3 Key Management

KM-1 The interface between the ATM Network Element and the Network Management System shall support a key management system for the secure distribution of key encryption keys that are shared between the Network Management System and the ATM Network Element.

KM-2 The interface between the ATM Network Element and the Network Management System shall support a key management system for the secure distribution of traffic protection keys that are shared between the Network Management System and the ATM Network Element.

KM-3 The key management scheme implemented shall provide perfect forward secrecy for all confidential communications between the ATM Network Element and the Network Management System.

2.4 Authentication

A-1 The interface between the ATM Network Element and the Network Management System shall authenticate all communications between the ATM Network Element and the Network Management System.

A-2 The interface between the ATM Network Element and the Network Management System shall support the capability mutually to establish and verify the claimed identity of the other entity.

2.5 Non-Repudiation

NR-1 The interface between the ATM Network Element and the Network Management System shall protect against any attempt by a message originator to deny sending a specific message.

NR-2 The interface between the ATM Network Element and the Network Management System shall protect against any attempt by a message recipient to deny receiving a specific message.

NR-3 The interface between the ATM Network Element and the Network Management System shall protect against any attempt by one party to deny that an authentication or session establishment protocol was run between two parties.

2.6 Access Control

AC-1 The ATM Network Element shall support the capability to limit the actions of an operator based upon the operator’s identity.

AC-2 The ATM Network Element shall support the capability to limit an operator’s privileges based on the method of access.

2.7 Audit

AU-1 The ATM Network Element shall be capable of recording a set of events that is specifiable by a Network Administrator.

AU-2 The ATM Network Element shall be capable of recording the system time at which each audited event occurred.

AU-3 The ATM Network Element shall be capable of recording the identity of the Network Administrator who performed each action.

2.8 Activity Reporting

AR-1 The ATM Network Element shall be capable of reporting events selected by a Network Administrator to the Network Management System as they occur in real time.

2.9 Security Recovery

SR-1 The ATM Network Element shall support recovery from incidents that impede or degrade the performance of the ATM Network Element.

3.0 Definitions and Acronyms

3.1 Definitions

In this specification, the following definitions apply:

ATM Network Element: Any device supporting one or more of the defined ATM interfaces or services. It may also support other interfaces or services.

Network Management System: The terminal, network element, or system that provides services to manage an ATM Network Element. It may manage multiple Network Elements, including non-ATM Network Elements.

Network Administrator: A person who is authorized to use a Network Management System.

3.2 Acronyms

The following acronyms or abbreviations are used in this specification:

AES	Advanced Encryption Standard
CBC	Cipher Block Chaining
CMIP	Common Management Information Protocol
CORBA	Common Object Request Broker Architecture
CRC	Cyclic Redundancy Check
DES	Data Encryption Standard
DH	Diffie-Hellman
DSS	Digital Signature Standard
IKE	Internet Key Exchange
IPSec	IP Security
KDC	Key Distribution Center
MAC	Message Authentication Code
MIB	Management Information Base
RADIUS	Remote Authentication Dial In User Service
RSA	Rivest, Shamir, and Adleman
SHA	Secure Hash Algorithm
SNMP	Simple Network Management Protocol
SSH	Secure Shell
SSL	Secure Sockets Layer
TCP	Transmission Control Protocol
TGT	Ticket Granting Ticket
TLS	Transport Layer Security

4.0 References

4.1 Normative References

The following references contain provisions that, through reference in this text, constitute provisions of this specification. At the time of publication, the editions indicated were valid. All references are subject to revision, and parties to agreements based on this specification are encouraged to investigate the possibility of applying the most recent editions of the references indicated below.

[1] Case, J., D. Harrington, R. Presuhn, and B. Wijnen, "Message Processing and Dispatching for the Simple Network Management Protocol (SNMP)," RFC 2272, Internet Engineering Task Force, January 1998.

[2] Dierks, T., and C. Allen, "The TLS Protocol," RFC 2246, Internet Engineering Task Force, January 1999.

[3] Freier, A.O., P. Carlton, and P.C. Kocher, "The SSL Protocol Version 3.0," http://home.netscape.com/eng/ssl3/draft302.txt, November 1996.

[4] Harrington, D., R. Presuhn, and B., Wijnen, "An Architecture for Describing SNMP Management Frameworks," RFC 2271, Internet Engineering Task Force, January 1998.

[5] Housley, R., W. Ford, W. Polk, and D. Solo, "Internet Public Key Infrastructure: Part I: X.509 Certificate and CRL Profile," RFC 2459, Internet Engineering Task Force, January 1999.

[6] Khare, R., and S. Lawrence, "Upgrading to TLS within HTTP/1.1," RFC 2817, Internet Engineering Task Force, May 2000.

[7] Kohl, J., and C. Neuman, "The Kerberos Network Authentication Service (V5)," RFC 1510, Internet Engineering Task Force, September 1993.

[8] Levi, D., P. Meyer, and B. Stewart, "SNMPv3 Applications," RFC 2273, Internet Engineering Task Force, January 1998.

[9] Medvinsky, A., and M. Hur, "Addition of Kerberos Cipher Suites to Transport Layer Security (TLS)," RFC 2712, Internet Engineering Task Force, October 1999.

[10] Rescorla, E., "HTTP over TLS," RFC 2818, Internet Engineering Task Force, May 2000.

[11] Rigney, C., S. Willens, A. Rubens, S. Willens, and W. Simpson, "Remote Authentication Dial In User Service (RADIUS)," RFC 2865, Internet Engineering Task Force, June 2000.

[12] Wijnen, B., R. Presuhn, and K. McCloghrie, "View-based Access Control Model (VACM) for the Simple Network Management Protocol (SNMP)," RFC 2275, Internet Engineering Task Force, January 1998.

[13] Wijnen, B., and U. Blumenthal, "User-based Security Model (USM) for version 3 of the Simple Network Management Protocol (SNMPv3)," RFC 2274, Internet Engineering Task Force, January 1998.

[14] Ylönen, T., "The SSH (Secure Shell) Remote Login Protocol," http://www.tigerlair.com/ssh/faq/ssh1-draft.txt, November 1995.

[15] Carasik, A., "Secure Shell FAQ," Revision 1.4, http://www.tigerlair.com/ssh/faq, February 2001.

4.2 Informative References

[16] Gutmann, P., "Software Generation of Practically Strong Random Numbers," *Seventh USENIX Security Symposium Proceedings*, The USENIX Association, 1998, pp. 243–257.

[17] Ylönen, T., "SSH—Secure Login Connections over the Internet," *Proceedings of the Sixth USENIX Security Symposium*, July 1996, pp. 37–42.

[18] Kelsey, J., B. Schneier, and N. Ferguson, "Notes on the Design and Analysis of the Yarrow Cryptographic Pseudorandom Number Generator," *Sixth Annual Workshop on Selected Areas in Cryptography*, Springer-Verlag, 1999.

[19] Rescorla, E., *SSL and TLS*, Addison-Wesley, 2001.

5.0 Command Line Interfaces

If an ATM Network Element provides for command line access, it shall support at least one of the following:

?? Kerberos,

?? RADIUS, or

?? SSH.

This requirement applies to *all* command line interfaces to the ATM Network Element. This includes management, administration, debugging ports, and any other interfaces not listed here.

5.1 Kerberos

Kerberos is a trusted-third party security system that uses a Key Distribution Center (KDC) to establish secure, authenticated sessions between a client and an application server. The contents of this section are aimed:

1. To promote conformance of management systems secured by Kerberos with the security objectives in Section 2, above,
2. To promote interoperability of such implementations with commonly available, current implementations, and
3. To help configure these systems according to generally accepted best practices.

Conformance with the methods described here satisfies security objectives C-1 and C-2. Individual implementations, independently, may or may not satisfy security objectives C-3, C-4, and C-5. Security objective C-6 is not satisfied.

Conformance with the methods described here satisfies security objectives I-1 and I-2. Individual implementations, independently, may or may not satisfy security objectives I-3 and I-4.

Conformance with the methods described here satisfies security objectives KM-1 and KM-2. Security objective KM-3 is not satisfied.

Conformance with the methods described here satisfies security objectives A-1 and A-2.

Security objectives NR-1, NR-2, and NR-3 are not satisfied.

Individual implementations, independently, may or may not satisfy security objectives AC-1, AC-2, AU-1, AU-2, AR-1, and SR-1. Use of TCP Wrappers may help to satisfy security objectives AC-1 and AC-2. Kerberos provides identification information needed to implement security objective AU-3.

A Network Element that supports Kerberos shall implement a Kerberos application server, which accepts Kerberos tickets and authenticators and uses these to provide two-way authentication and to support additional security services. These implementations shall support the Kerberos Version 5 Specification 1 (5.1) as defined in [7] (see, especially, Section 9.1). In addition:

?? The KDC shall be physically secured and should be run on a stand-alone processor with no other applications. It should have no other users besides Kerberos administrators, and it should have all other network services disabled (except for logging, auditing, backup, or intrusion detection).

?? The KDC shall turn on Kerberos's auditing.

?? The KDC shall use TCP port 88.

?? The KDC should set all client principals to expire once a year.

?? The KDC must use a cryptographically strong method of generating random or psuedo-random numbers. See [16] and [18] for additional guidelines and recommendations.

?? Cross-realm operation should be avoided. If cross-realm operation is used, cross-realm authentication shall be direct.

?? TGTs shall be issued for at most 8 hours and be renewable for at most 7 days.

?? Allowable clock skew shall be no more than 5 minutes, and application servers shall maintain a replay cache of at least 10 minutes.

?? Application servers shall use random passwords and store encrypted passwords in restricted access or otherwise protected files. Application servers should not be allowed to obtain tickets.

?? Clients should not use random passwords, unless the clients themselves are implemented as automated scripts, in which case they should use random passwords and protect these passwords the same way application servers do.

?? All principals should change passwords every 3 to 6 months.

?? Clients and application servers shall support "kerberized" telnet, shall support "kerberized" ftp, and may support "kerberized" rsh or rlogin.

?? Clients' tickets should not be "forwardable" and not "proxiable".

?? Except for use in automated scripts, tickets shall not be post dated.

?? Sessions between clients and application servers shall use two-way authentication (KRB_AP_REQ MUTUAL REQURED), shall use integrity protection (KRB_SAFE), and may use confidentiality (KRB_PRIV).

?? For encryption, algorithms, implementations shall support DES-CBC, should support 3-DES-CBC, and should support AES when available.

?? For authentication algorithms, implementations shall support MD5, should support SHA-1, may support DES-MAC or DES-MAC-K, shall not use CRC 32, and shall not support MD4.

The following items refer to features currently being considered for IETF standards. If future standards specify such functionality, then:

?? Encryption with AES should be supported.

?? Clients may be required to use the hardware authentication function.

?? Public key initial authentication (PK-INIT) should be supported.

5.2 RADIUS

A Network Element that supports RADIUS shall implement a RADIUS client to authenticate a user from a RADIUS server and shall use that authentication as the sole authentication of the client. These implementations shall support RADIUS as defined in [11].

RADIUS is a protocol designed to perform authentication, authorization, and accounting. It is not designed to provide confidentiality, integrity, or any key management services. If these security services are needed, users may consider deploying RADIUS over an IPSec tunnel or other comparable solutions.

RADIUS may be used with PAP, CHAP, UNIX login, or any other authentication mechanism. When used with PAP, RADIUS protects the PAP ID and password with a shared secret.

RADIUS specifies client-to-server authentication and does not specify a server-to-client authentication mechanism. RADIUS does not specify a user-to-client authentication mechanism.

RADIUS uses a shared secret between the client and server. It does not specify how to establish or change this shared secret. If RADIUS proxy servers are used, the secret must also be shared with any participating proxy servers.

The following three recommendations are contained in [11]:

?? RADIUS implementations are cautioned against using keep alives.

?? RADIUS implementations should use the officially assigned UDP port of 1812.

?? RADIUS uses a challenge response mechanism. The server sends a challenge message to the client consisting of a random number. The client shall encrypt the random number using the shared secret and return it to the server. The random number should be at least 16 octets. Implementations shall have access to a source of cryptographically strong random or pseudo-random numbers. See [16] and [18] for additional guidelines and recommendations.

RADIUS was designed as an authentication protocol and consequently, security objectives C-1, C-2, C-3, C-4, C-5, C-6, I-1, I-2, I-3, KM-1, KM-2, KM-3, NR-1, NR-2, and NR-3 are not met by RADIUS. Individual implementations, independently, may or may not satisfy security objectives I-4, AC-1, AC-2, AU-1, AU-2, AR-1, and SR-1.

Security objective A-1 is fully met by RADIUS.

RADIUS is designed to authenticate the client to the server. RADIUS does not authenticate the server to the client. Therefore, security objective A-2 is not met by RADIUS.

RADIUS can be configured to accept a valid user password protected by a shared secret, or to require a challenge and response in addition to a valid protected password. RADIUS implementations should require clients to process a challenge and response.

5.3 SSH

The Secure Shell (SSH[1]) defines security protocols that use public key cryptography to establish secure, authenticated sessions between a client and a server. The contents of this section are aimed:

?? To describe how management systems secured by SSH conform with the security objectives in Section 2, above,

?? To promote interoperability of such implementations with commonly available, current implementations, and

?? To help configure these systems according to generally accepted best practices.

Conformance with the methods described here satisfies security objectives C-1 and C-2. Individual implementations, independently, may or may not satisfy security objectives C-3, C-4, and C-5. Security objective C-6 is not satisfied.

Conformance with the methods described here satisfies security objectives I-1 and I-2. Individual implementations, independently, may or may not satisfy security objectives I-3 and I-4.

Conformance with the methods described here satisfies security objectives KM-1 and KM-2. Security objective KM-3 may be partially satisfied when using the SSH1 protocol.

Conformance with the methods described here satisfies security objectives A-1 and A-2.

Security objectives NR-1 NR-2 are not satisfied. Support for satisfying security objective NR-3 in one direction, however, is provided if the client uses the public key authentication method.

Individual implementations, independently, may or may not satisfy security objectives AC-1, AC-2, AU-1, AU-2, AR-1, and SR-1. Using TCP Wrappers may help to satisfy security objectives AC-1 and AC-2.

5.3.1 Description of SSH

SSH1 [14], [17] and SSH2 [15] are two completely distinct protocols. Both have freely available specifications and have been implemented in freeware and commercial products. Neither is a standard [as of November 2001], although SSH2 was described at the time in several Internet Drafts. Because SSH1 and SSH2 servers bind to the same TCP port, and the protocol begins with an exchange of protocol and software version numbers, it is possible for a SSH2 server to launch a SSH1 server to handle a SSH1 client.

SSH is intended to allow a user to logon, execute commands, or transfer files securely. It is a replacement for telnet, rlogin, rsh, and rcp. It provides strong authentication and secure communications. An integrated “port forwarding” feature can be used to secure X11 connections or in fact any TCP connection, e.g., to perform a secure remote backup. SSH2 has an explicit capability to secure ftp as well.

A description of SSH begins with the transport layer protocol. In SSH1, two levels of public keys are used. A client sends an authentication request to a server, and the server responds with its

[1] SSH is a registered trademark and Secure Shell is a trademark of SSH Communications Security Ltd. of Finland.

long-term public host key and public server key (which changes hourly). In SSH2, only the host key is present. The client compares the host key, which in the former case authenticates the server key, with that which has already been configured. A client may be configured to trust new host keys or not. Note that certificates are not used directly by SSH, but a use of PKI may be added in the future. To make sure that these first two messages of the key exchange sequence itself have not been manipulated, both parties compute a hash of the initial messages and session key, which they use later as a session identifier.

After the client receives and verifies the server's public key(s), it chooses a 256-bit pseudorandom number, which becomes the basic shared secret from which all uni-directional session keys are derived. The random number, a known constant, and the session identifier are double encrypted with the server and host keys in SSH1 or singly encrypted with the host key in SSH2. This value is returned along with a choice of traffic protection algorithms. In SSH1, this provides perfect forward secrecy for the traffic confidentiality keys with respect to the host key.

SSH provides for the negotiation of both traffic protection and compression algorithms. SHA-1 and 3-DES are mandatory to implement, but other popular choices as well as proprietary algorithms can also be used. A reliable transport stream in each direction (i.e., TCP) is required, and packet sequencing is additionally verified by including an implicit sequence number in each MAC calculation. Either party may request rekeying at any time.

The SSH authentication protocol is layered on top of the transport layer protocol. The next step is user authentication, which can be done with a password over the secure channel, token-based systems, or the user's public-private key pair. In the last of these cases, a pass-phrase protects the user's private key on the client system. (SSH1 also supports Kerberos for user authentication.) After the authentication protocol completes successfully, the client may request different protected services from a list of supported services. These services are then protected with SSH encryption, MACs, and secured end of file messages.

5.3.2 Use of SSH

The following guidelines are provided for the use of SSH to protect the management of an ATM Network Element.

?? Official releases of the software from SSH Communication Security are signed. Implementers or users downloading these releases of SSH should verify these signatures.

?? SSH2 contains improvements in performance, security, and portability over SSH1. In particular, certain active attacks against the SSH1 protocol are prevented in SSH2. Therefore, client and server implementations should support SSH2.

?? Implementations of SSH clients and servers must use a cryptographically strong method of generating pseudo-random numbers. See [16] and [18] for additional guidelines and recommendations.

?? Deployments of SSH should use public key authentication. Deployments may use passwords or, in the case of SSH1, Kerberos. Host-based authentication should not be used.

?? Client computers must be protected from attempts to modify their configured host keys or to obtain their private keys. Such protection includes physical access to and modification of the software, as well as other compromises.

?? Clients must not accept new, unconfigured host keys for access to ATM Network Elements.

?? SSH servers must be protected so that host private keys are not revealed and, in the case of public key authentication, users' public keys are not altered. If passwords or another type of authentication is used, such authentication data must also be protected appropriately to avoid both direct attacks and dictionary attacks.

?? The server (sshd) should be run directly and not from inetd. It may be configured with TCP Wrappers.

?? SSH should not be configured with public key sizes shorter than 768 bits.

?? If an ATM Network Element runs a SSH server, it may be configured with a SSH client as well.

6.0 Securing Web-Based Management

If a network element provides for web-based management, it shall support at least one of the following:

?? SSL, or

?? TLS.

6.1 General Requirements

The Secure Sockets Layer (SSL) and Transport Layer Security (TLS) protocols provide cryptographic authentication, data stream integrity, and data stream confidentiality for TCP connections. They are particularly well suited for protecting http traffic between web browsers and servers, but they may be used to protect any protocol running over TCP (e.g., telnet, rlogin, or even SNMP). The contents of this section are aimed to promote conformance of web-based management systems with the security objectives in Section 2, to promote interoperability of such implementations with commonly available, up-to-date software, and to help configure these systems according to generally accepted best practices.

Conformance with the methods described here satisfies security objectives C-1 and C-2. Individual implementations, independently, may or may not satisfy security objectives C-3, C-4, and C-5. Security objective C-6 is not satisfied.

Conformance with the methods described here satisfies security objective I-1. When used together with TCP, security objective I-2 is satisfied. Individual implementations, independently, may or may not satisfy security objectives I-3 and I-4.

Conformance with the methods described here satisfies security objectives KM-1 and KM-2. Security objective KM-3 is not satisfied.

Conformance with the methods described here satisfies security objectives A-1 and A-2. Using TCP Wrappers may help to satisfy security objectives AC-1 and AC-2.

Security objectives NR-1 and NR-2 are not satisfied. Support for satisfying security objective NR-3 in one direction, however, is provided if the client uses certificate-based authentication.

Individual implementations, independently, may or may not satisfy security objectives AC-1, AC-2, AU-1, AU-2, AU-3, AR-1, and SR-1. Use of TCP Wrappers at the server may help to satisfy security objectives AC-1 and AC-2.

WBM-1: An ATM Network Element that provides an HTTP server shall support SSLv3 with RSA [3]. It may also support SSLv3 with DSS and DH or TLS 1.0 [2].

Other protocols (e.g., SSLv2, PCT, and S-HTTP) are outside the scope of this document.

For certain e-commerce applications, the burden of authentication is placed on the server, because the browser can supply the required payment credentials like credit card data when needed. For applications like network management, authentication of both parties is critical. The recommended method is to outfit both parties with certificates signed by the network operator, install the network operator's public key as the root key in the browsers and servers, and remove all other trusted root keys from the server and the browser. That is, both parties (when using RSA, for example) respond to the CertificateRequest message with a Certificate message and a

CertificateVerify message. A less secure alternative method of client authentication is to use a hardware-token-based one-time password system over the secured connection. Simple passwords sent over the secure connection may be vulnerable to a number of practical attacks, so these should be used only with carefully constructed constraints (aging, complexity, logging, protection against dictionary attacks, etc.).

WBM-2: Clients (browsers) should use certificates to authenticate to the server. They may, however, use a token-based authentication system or passwords sent over the protected channel.

WBM-3: Certificates should be generated with a lifetime of no more than two years, and entire certificate chains shall be checked for correct names and expiration and should be checked for revocation.

WBM-4: Both parties shall have access to a source of cryptographically strong random or pseudo-random numbers. See [16] and [18] for additional guidelines and recommendations.

WBM-5: The server shall support RSA; it may support DH-DSS; it may support Kerberos as described in [9]; and it may support the Fortezza cipher suites, but see [19] for a discussion of problems using Fortezza as described in [3]. For RSA or DH-DSS, key lengths shall be at least 768 bits and both servers and browsers should support 1024-bit keys. The same goes for all certificates in the chain. Applications requiring confidentiality should use 3-DES or RC4-128. AES should be supported when it becomes available. Proprietary cipher suites may also be used.

WBM-6: Both parties shall provide long-term protection for the privacy of their authentication data and the integrity of root public keys they rely upon to verify certificates. Hardware tamper resistance like a smart card or cryptographic module is preferable to disk storage, but if disk storage is used, these items should be encrypted and password protected, and the system should log all attempted accesses securely. Off-line storage is preferable to on-line storage for these long-term keys.

WBM-7: Both parties shall protect pre-master secrets, master secrets, and session keys for the duration of their use. Use of software that allows unrestricted access to main memory, memory dumps, examination of paging devices, and so forth shall be restricted accordingly. Processes should be locked in main memory and not paged where practical.

WBM-8: Session resumption with a timeout may be used. The recommended timeout interval is ten minutes.

6.2 SSLv3

The SSLv3 protocol (protocol version major=3, minor=0) is specified in [3], which can be found on Netscape’s Web site.

WBM-9: Port and protocol selection and use shall follow [10].

WBM-10: The ephemeral RSA, anonymous, and Server Gated Cryptography options shall not be used.

WBM-11: The server shall use the close_notify alert. The browser should also use close_notify to complete a two-way closure handshake.

WBM-12: Both parties should support protected Rehandshake exchanges.

6.3 TLS 1.0

A Network Element that supports the TLS protocol (protocol version is major=3, minor=1) shall support it as defined in [2].

WBM-13: If TLS 1.0 is supported, the requirements for connection closure, use of port numbers, checking the server's identity, and checking the client's identity in [10] shall be followed.

WBM-14: If TLS 1.0 is supported, the name matching rules specified in [5] shall be followed.

WBM-15: Servers should and browsers may support the use of port numbers described in [6].

6.4 Securing the Browser

In general, new browsers (released in 2000 or later) should be preferred over older ones, because older protocols like SSLv2 have security defects, and cryptographic strength has increased since the easing of U.S. export restrictions in January 2000. For the same reason, U.S. export-only versions should be avoided.

WBM-16: The browser should be configured with its security settings to support the requirements listed above. Features that are not used, such as plug-ins, Java, JavaScript, or ActiveX controls, should be disabled. Unneeded CAs' certificates should be removed. The browser and the platform on which it is running should be isolated from the possibility of unauthorized modification. Extraneous network services should be disabled. System logging and intrusion detection tools should be used to monitor the configuration as appropriate.

WBM-17: The browser should wait for the server's handshake Finish message before sending application data.

7.0 MIB–Based Management

A Network Element that supports a MIB-based network management interfaces shall support SNMP.

7.1 SNMP

A Network Element that supports an SNMP access over protocols other than TCP shall support SNMPv3 as defined in [4], [1], [8], [13], and [12]. A Network Element that supports SNMP access over TCP shall support either SNMPv3 as described in this section or SSL-TLS as described in Section 6. If SNMP is run over TCP, then use of TCP Wrappers may help to satisfy security objectives AC-1 and AC-2. Otherwise, the remainder of this section applies to the use of SNMPv3. Individual SNMPv3 implementations, independently, may or may not satisfy security objectives AC-2, AU-1, AU-2, AR-1, and SR-1. SNMPv3 supports a view based access control model that fully supports security objective AC-1.

7.1.1 Key Management

SNMPv3 contains no provision for security association negotiation or session key generation. SNMPv3 assumes that the caller will select the proper key to use for each service and will somehow pass the key in a secure manner to all SNMP engines that require the key.

Implementations of SNMPv3 with preplaced initial keys and the rekey option minimally satisfy KM-2. Using IKE or Kerberos satisfies KM-2. Entities implementing the rekey option shall have access to a source of cryptographically strong random or pseudo-random numbers. See [16] and [18] for additional guidelines and recommendations.

To satisfy KM-1, implementations of SNMPv3 shall use some other secure key distribution protocol (e.g., Kerberos or IKE).

To satisfy KM-3, implementations of SNMPv3 shall use some other secure key distribution protocol (e.g., certain options of IKE).

The key localization algorithm transforms the user's password into a traffic encryption key shared between a user and one authoritative SNMP engine. Implementations of SNMPv3 using a key management scheme shall not use the key localization algorithm option.

7.1.2 Proxy

Many SNMP implementations make use of proxy agents. SNMPv3 specifies that a proxy forwarding application, "must perform a translation of incoming management target information into outgoing management target information. How this translation is performed is implementation specific." This implies that proxy agents shall have access to the SNMP packets. Therefore, the proxy agents need to have access to privacy keys and authentication keys. Security between a management station and an end device may have several proxies processing plaintext messages in the path. In fact a secure message from the management station may be translated into an insecure message by any proxy agent in the path.

SNMPv3 does not protect against denial of service or traffic analysis.

7.1.3 Integrity

The SNMPv3 specification lists HMAC-MD5-96 as “shall support” and HMAC-SHA-96 as “should support” for authentication and data integrity. Either algorithm satisfies security objective I-1. SNMPv3 satisfies security objective I-2 when used in conjunction with the timeliness option.

Security objectives I-3 and I-4 are not met by SNMPv3.

7.1.4 Timeliness

SNMPv3 provides a timeliness feature only if authentication is used. SNMPv3 specifies a time window of 150 seconds in which SNMP messages shall be received from the time they are sent.

Timeliness should be used by SNMPv3 implementations, which satisfies security objective I-2 when used in conjunction with integrity.

7.1.5 Confidentiality

The SNMPv3 specification names DES-CBC as the only privacy algorithm. DES-CBC may not be appropriate for securing some management applications.

Implementations of SNMPv3 using DES-CBC minimally satisfy security objective C-1. Implementers are urged to use stronger alternatives.

Security objectives C-2, C-3, C-4, C-5 and C-6 are not met by SNMPv3.

Although there is a rudimentary mechanism for rekeying, it is not considered adequate to satisfy security objective C-2.

Security objectives A-1 and A-2 are fully met by SNMPv3.

Security objectives NR-1, NR-2 and NR-3 are not met by SNMPv3, but security objective NR-3 may be satisfied fully or partially, by certain key management protocols used with SNMPv3 (e.g., IKE).