Although SMTP was designed as a mail transport and delivery protocol, this specification also contains information that is important to its use as a “mail submission” protocol for “split-UA” (User Agent) mail reading systems and mobile environments. 6. Secure/Multipurpose Internet Mail Extensions (S/MIME), Version 3.2, Message Specification, Proposed Standard, IETF RFC 5751, ISSN: 2070-1721, Ramsdell and Turner, January 2010. https://datatracker.ietf.org/doc/rfc5751. NIST SP 1800-6B: Domain Name System-Based Electronic Mail Security 31 This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1800-6. RFC 5751 defines S/MIME version 3.2. S/MIME provides a consistent way to send and receive secure MIME data. The RFC describes methods for digital signatures to provide authentication, message integrity, and non-repudiation with proof of origin; encryption to provide data confidentiality; and to reduce data size. 7. Use Cases and Requirements for DNS-Based Authentication of Named Entities (DANE), IETF RFC 6394, ISSN: 2070-1721, Barnes, October 2011. https://datatracker.ietf.org/doc/rfc6394. Many current applications use the certificate-based authentication features in TLS to allow clients to verify that a connected server properly represents a desired domain name. Typically, this authentication has been based on PKI certificate chains rooted in well-known certificate authorities (CAs), but additional information can be provided via the DNS itself. This document describes a set of use cases in which the DNS and DNSSEC could be used to make assertions that support the TLS authentication process. The main focus of this document is TLS server authentication, but it also covers TLS client authentication for applications where TLS clients are identified by domain names. 8. The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security Protocol: TLSA, Proposed Standard, IETF RFC 6698, ISSN: 2070-1721, Hoffman and Schlyter, August 2012. https://datatracker.ietf.org/doc/rfc6698. Encrypted communication on the Internet often uses TLS, which depends on third parties to certify the keys used. RFC 6698 provides means to improve on that situation by standardizing on methods to enable the administrators of domain names to specify the keys used in that domain’s TLS servers. This requires matching improvements in TLS client software, but no change in TLS server software. 9. Updates to the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, Proposed Standard, IETF RFC 6818, ISSN: 2070- 1721, Yee, January 2013. https://datatracker.ietf.org/doc/rfc6818 RFC 6818 updates RFC 5280, the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile. It changes the set of acceptable encoding methods for the explicit Text field of the user notice policy qualifier and clarifies the rules for converting internationalized name labels to American Standard Code for Information Interchange (ASCII). The RFC also provides some clarifications on the use of self-signed certificates, trust anchors, and some updated security considerations. 10. SMTP security via opportunistic DANE TLS, RFC 7672, Dukhovni and Hardaker, May 26, 2015. https://datatracker.ietf.org/doc/rfc7672 The RFC describes a downgrade-resistant protocol for SMTP transport security between Message Transfer Agents, based on the DANE TLSA DNS record. Adoption of this protocol will NIST SP 1800-6B: Domain Name System-Based Electronic Mail Security 32 This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1800-6. enable an incremental transition of the Internet email backbone to one using encrypted and authenticated TLS. 11. Using Secure DNS to Associate Certificates with Domain Names For S/MIME, RFC 8162, Hoffman and Schlyter, May 2017. https://datatracker.ietf.org/doc/rfc8162/ The draft RFC for using secure DNS to associate certificates with domain names for S/MIME describes how to use secure DNS to associate an S/MIME user’s certificate with the intended domain name; similar to the way that DANE (RFC 6698) does for TLS. 3.5 Technologies The laboratory configuration employed for the project included components contributed by several sets of collaborating organizations. One of the component sets is Windows-based. The others are Linuxbased. There were also three MUAs: Microsoft Outlook, Mozilla Thunderbird (on Linux), and a Thunderbird MUA equipped with a DANE-aware Apple Key Chain utility that were able to interact with all the mail servers via IMAP. While the Windows-based contribution used Server 2016 DNS services, the Linux-based contributions included three different implementations for DNS. One was based on NSD4 and Unbound authoritative and recursive servers, one was based on the Berkeley Internet Name Domain (BIND) DNS server, and one was based on the Secure64 DNS services. Secure 64 also contributed DNS services hosted on dedicated processors using SecureT micro operating system (OS) technology. Collaborators assisted in installation and initial configuration of products and, as necessary, in composition of components for different test cases. Figure 3.1 below depicts, at a high level, collaborator contributions used to support the demonstration project. Elements identified in boldface are components provided or adapted by the collaborator. Other elements were incorporated into the stack to permit checking out the installed component’s functionality. Collaborator contributions identified below are organized with respect to the contributor as initially installed and checked out at the NCCoE. The architecture described in Section 4 below permits demonstration of the interconnection of components provided by different collaborators and initially checked out independently. NIST SP 1800-6B: Domain Name System-Based Electronic Mail Security 33 This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1800-6. 3.5.1 Microsoft The Microsoft environments were contributed to support demonstration Scenario 1. Two environments were configured on the laboratory’s VMware virtual machines (see Figure 3.1 above). Each stack included the ability to demonstrate Office Outlook30 as an MUA, included Exchange Server 201631 as MTAs, and used Active Directory running on Microsoft Windows Server 201632 for DNS services. The Microsoft contribution included a DNSSEC-aware DNS recursive server, a DNSSEC-aware DNS authoritative server (IETF RFC 4033, 4034, and 4035), an MTA that can do SMTP over TLS (RFC 3207), management tools to configure servers and for debugging purposes, X.509 certificate sources, FIPS 140- 30 https://en.wikipedia.org/wiki/Microsoft\_Outlook 31 https://products.office.com/en/exchange/microsoft-exchange-server-2016 32 https://www.microsoft.com/en-us/cloud-platform/windows-server Figure 3.1 DNS-Based Email Security Collaborator Contributions NIST SP 1800-6B: Domain Name System-Based Electronic Mail Security 34 This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1800-6. 2 validated cryptographic software, and support for multifactor authentication. The stacks were also able to be configured to demonstrate that Exchange could be used with either an Outlook or a Thunderbird MUA. Other test cases were demonstrated using Exchange with a combination of other providers’ DNS implementations. 3.5.2 NLnet Labs The NLnet Labs contribution focused on DNS services to support both demonstration scenarios. NLnet software was initially configured on the laboratory’s VMware virtual machines. The components included NSD4 4.1.933, Unbound34, and OpenDNSSEC35 software for DNS services and Postfix and Dovecot for mail services. NSD4 is an authoritative only, high performance, open source name server. Unbound is a validating, recursive, caching DNS resolver. OpenDNSSEC is a set of software for signing DNS zones that are then served using NSD. While OpenDNSSEC can be configured to sign zone files or to sign zones transferred in via DNS zone transfer (AXFR), in these scenarios, it is used to sign local zone files in these scenarios. Like with the Microsoft stack above, multiple MUAs were configured to send and receive mail with the NLnet components via SMTP and IMAP. 3.5.3 Internet Systems Consortium (ISC) The ISC contribution was focused on the BIND DNS server and supported both demonstration scenarios. BIND was initially configured on the laboratory’s VMware virtual machines and included configuration for Postfix and Dovecot for email. BIND21 is open source software that is considered the reference implementation of DNS, but it is also production-grade software, suitable for use in high-volume and high-reliability applications. BIND features response rate limiting (RRL), support for FIPS 140-2 validated hardware cryptographic modules, the optional ability to retrieve zone data directly from an external database, the ability to use inline signing to automatically re-sign records as they are updated, and a scalable master/slave hierarchy. Like the other stacks, all three MUAs were able to connect and use the stack for DNS and email. BIND versions prior to BIND 9.11.0 are released under the ISC License (https://www.isc.org/downloads/software-support-policy/isc-license/). 3.5.4 Secure64 The Secure64 contributions were focused on DNS services to support both demonstration scenarios. The Secure64 environment included an automated online Secure64 DNS Signer as well as DNSSECcapable VM images of DNS Cache, DNS Authority, and DNS Manager. DNS Manager provided centralized management of Secure64 DNS Cache software and configurations and provided network-wide monitoring of key performance indicators. DNS Manager allowed creation of groups of servers and 33 https://www.nlnetlabs.nl/projects/nsd/ 34 http://unbound.net 35 https://www.opendnssec.org NIST SP 1800-6B: Domain Name System-Based Electronic Mail Security 35 This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1800-6. assignment of configurations to a group, a single server, or all servers. DNS Authority is an authoritative signer and server as a single platform. DNS Cache, DNS Authority, and DNS Manager were configured on the laboratory’s VMware virtual machine; and the DNS Signer was provided as a high-assurance implementation delivered on a Secure64 dedicated appliance. Secure64 contributions were able to demonstrate Outlook, Thunderbird, or Thunderbird equipped with an Apple Key Chain utility as MUAs and use Postfix as an MTA and Dovecot to provide IMAP for clients. 4 Architecture The Security platform architecture used for the project included combinations of components from different sources that supported two usage scenarios for DANE-enabled secure email in four different systems environments. 4.1 Usage Scenarios Supported The scenarios supported include: ♣ “ordinary” email where the email exchanges between two organizations’ email servers communicate over TLS with a STARTTLS extension, and relevant TLSA records are published in the receiver’s DNS zone protected by DNSSEC ♣ end-to-end signed email, where the email exchanges between users in different organizations are carried over a channel protected by TLS (using the STARTTLS extension), and relevant artifacts used for signing and channel protection are published in a DNS zone protected by DNSSEC. Subsequently, these artifacts are used for S/MIME and TLS validation. In both scenarios, end-entity and personal certificates were generated from CAs. Use of “well known” (i.e., installed as trust anchors in hosts), local enterprise CAs, and self-signed certificates were demonstrated. While the second scenario demonstrated signing of emails, it does not include an end-to-end encrypted email scenario. Signing addresses the main security concerns in enterprise environments, which are the target of the project, but may neglect concerns of individual users who may also want to reduce information disclosure to their email providers. The two scenarios that are included may, however, serve as enablers for end-to-end encryption. Participation by parties having a primarily end-to-end encryption focus may succeed in generating industry support for the building blocks needed to support end-to-end encryption. In more detail, the project’s security platforms use the STARTTLS extension to include encryption of communications between two MTAs, as well as the signature of individual messages using S/MIME. The encryption and decryption with S/MIME on the end user’s client was excluded from the current platform demonstration. NIST SP 1800-6B: Domain Name System-Based Electronic Mail Security 36 This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1800-6. 4.1.1 Usage Scenario 1 An individual needs to enter into an email exchange with an individual in another organization. Each individual exchanges email via the respective parent organization’s mail servers. Users connect to their organization’s respective mail servers within a physically protected zone of control. In this scenario, the privacy policy of the parent organization requires encryption of the information being exchanged. The security afforded by the cryptographic process is dependent on the confidentiality of encryption keys. The mail servers are configured to use X.509 certificates to authenticate themselves during an encryption key establishment process. DNSSEC is employed to ensure that each sending mail server connects to the legitimate and authorized receiving mail server from which its X.509 certificate is obtained. DANE resource records are employed to bind the cryptographic keying material to the appropriate server name. STARTTLS is employed to negotiate the cryptographic algorithm to be employed with TLS in the email exchange in which the PII is transferred. Encryption of the email message is accomplished by the originator’s email server, and decryption of the email message is accomplished by the recipient’s email server. Demonstrations of the security platform in this scenario include an attempt by a fraudulent mail server to pose as the legitimate receiver of the email and a man-in-the-middle attacker to attempt to disrupt the signal that TLS is available for the desired destination. In the latter attack, the goal is to force unencrypted transmission of the email. Both attempts should fail due to use of DNSSEC and DANE. 4.1.2 Usage Scenario 2 An individual needs to enter into an email exchange with an individual in another organization. Each individual exchanges email via the respective parent organization’s mail servers. Users connect to their organization’s respective mail servers within a physically protected zone of control. The policy of the parent organization requires cryptographic digital signature of the message to provide integrity protection source authentication of the email message. S/MIME is a widely available and used protocol for digitally signing email. Each organization has therefore generated X.509 certificates for their users that include the public portion of their signature keys. These certificates are then published in the DNS using the appropriate DANE DNS Resource Record (RR) type. DNSSEC is used to provide assurance that the originating user’s mail server connects to the intended recipient’s mail server. DANE records are employed to bind the cryptographic certificates to the appropriate server (for TLS) and individual user (for S/MIME), respectively. TLS is employed to provide confidentiality. Digital signature of the email message is accomplished by the originator’s email client. Validating the signature (hence the integrity of the authorization provided in the email message) is accomplished by the recipient’s email client. NIST SP 1800-6B: Domain Name System-Based Electronic Mail Security 37 This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1800-6. Demonstrations of the security platform in this scenario include an attempt by a fraudulent actor to pose as the originator of the email and a man-in-the-middle attacker attempting to disrupt the validation of the S/MIME signature. Both attempts fail due to use of DNSSEC and DANE records. 4.2 Architectural Overview The laboratory architecture for the project was designed to permit interconnection of Microsoft Outlook and Thunderbird MUAs with Microsoft Exchange and Postfix/Dovecot MTAs. It demonstrates the interconnection of either MTA with any of the DNS services contributed by collaborators. Two instantiations of each MTA type were established to demonstrate email exchanges between MTAs of the same type or different types. The various component combinations are then demonstrated with three different TLSA RR parameters: a self-signed certificate, use of local certificate authorities, and use of well-known certificate authorities. Figure 4.1 is a deployment diagram of the architecture used for demonstrating DNS-based email security. NIST SP 1800-6B: Domain Name System-Based Electronic Mail Security 38 This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1800-6. Figure 4.1 DNS-Based Email Security Deployment Diagram For test documentation purposes, the receiving MTA is named differently depending on the receiver’s DNS service zone and the TLSA option being demonstrated. The sending MTA’s implementation and DNS infrastructure can also vary for each test, but share the same basic processes. The design of the environment permits interconnection of components provided by different collaborators (see Figure 4.2). NIST SP 1800-6B: Domain Name System-Based Electronic Mail Security 39 This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1800-6. Figure 4.2 DNS-Based Email Security Test Setup The depiction shows that the project security platform test/demonstration activity was based on three different clients, two MTAs, and four DNS service configurations in the lab at the NCCoE exchanging messages with NLnet Labs and Secure64. All messages were signed (a mail client function) and encrypted (server to server). We worked with one remote location at a time, driven by whichever is ready first. The message exchanges, including DNS activity will be logged at each end (lab and remote correspondent). The solid connectors in the depiction illustrate one case. The dotted lines depict the other cases we want to demonstrate. A switch convention is used to reflect configuration options, but the project team actually configures each component for each option. The orange arrows between the mail clients and the Postfix MTA reflect the fact that clients submitted email directly to the SMTP server for relay, while using Dovecot only to get mail. (The depiction in NIST SP 1800-6B: Domain Name System-Based Electronic Mail Security 40 This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1800-6. Figure 4.2 reflects that IMAP is not used to submit mail, only retrieve it, so the MUA sent mail directly to the Postfix server, but received the reply through the Dovecot server.) The project team demonstrated 30 different events using various combinations of MUA, MTA, and DNS Server components divided among five test sequences. In each sequence, signed and encrypted messages were sent from a sender to a recipient. Both Exchange and Postfix encrypted mail by default. Most of the exchanges employed either self-signed certificates or local CAs (see Appendix C). The BIND configuration was set up to obtain and validate certificates from the NIST Advanced Network Technologies Division’s (ANTD’s) DNS source (acting as a root CA). (See section 6 below for test sequence sets.) In one test sequence, fraudulently signed S/MIME email was sent from a malicious sender to recipients using Outlook and Thunderbird MUAs configured to use Exchange and Postfix as MTAs. The Outlook/Exchange configuration used Active Directory as its DNS server. The configurations employing Postfix/Dovecot MTAs were demonstrated with each of the other three contributed DNS services. In one event, the Thunderbird MUA employed an Apple Key Chain Utility tool that allows a host to obtain X.509 certificates via of DANE RRs. All events were conducted using well-known CA and Enterprise CAissued certificates for the impersonated sender. The fraudulent site attempted to spoof a valid sending domain belonging to a Secure64 site that was configured with DNS Authority/Cache/Signer DNS services, a Postfix/Dovecot MTA, and Thunderbird equipped with the Apple Key Chain utility. An Outlook/Exchange/ Active Directory set-up acted as the fraudulent site. The email exchange between organizations was carried over TLS, and the email message was S/MIME signed on the fraudulent user’s client device. The setup for this sequence is depicted in Figure 4.3 below. In another sequence, an NCCoE system attempted to send a TLS protected email from Exchange and Postfix MTAs (in turn) to an external Postfix MTA using DNS Authority/Cache/Signer for DNS services. The NCCoE Exchange MTA used Active Directory DNS Services, and the Postfix/Dovecot MTA used BIND and NSD4/Unbound/OpenDNSSEC DNS services. An S/MIME signed email was sent to an external Postfix MTA. Four events were conducted using Well-Known CA issued certificates, four events were conducted using Enterprise CA issued certificates (TLSA/SMIMEA RR parameter of CU=2) for TLS and S/MIME on the receiver side, and three events were conducted using self-signed certificates (TLSA/SMIMEA RR parameter of CU=3) for TLS and S/MIME on the receiver side. An Outlook/Exchange/Active Directory stack acted as a man-in-the-middle and attempted to intercept the message. Figure 4.4 depicts the configuration for a man-in-the-middle demonstration. Note that the sender is being misdirected to a malicious email server only. This is to simulate a lower-level attack where email is sent (via route hijacking or similar low-level attack) to a man-in-the-middle. Figure 4.4 depicts the configurations used with the Thunderbird/Postfix/Dovecot/Bind option selected. NIST SP 1800 -6B: Domain Name System -Based Electronic Mail Security 41 This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1800-6. Figure 4 . 3 Fraudulent DNS Address Spoofing Configurations NIST SP 1800-6B: Domain Name System-Based Electronic Mail Security 42 This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1800-6. Figure 4.4 Man-In-The-Middle Event Configurations The following subsections describe the architecture’s MUA, MTA, and DNS service components and Cybersecurity Framework Core Subcategories supported by those components. 4.2.1 Client Systems and MUAs Client systems environments are Microsoft Office, Apple Mail, and open-source Linux-based Thunderbird applications. These include both commercial products and open-source software. MUA capabilities associated with the client systems are used to invoke S/MIME digital signature and signature verification for email, but user-to-user encryption is not demonstrated. Collaborators assisted in installation, integration tailoring as necessary, and testing of laboratory configurations. NIST SP 1800-6B: Domain Name System-Based Electronic Mail Security 43 This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1800-6. Table 4.1 Client Systems Application Source Collaborator Configuration Support Cybersecurity Framework Category Office Outlook Mail User Agent Microsoft Microsoft PR.AC-1, PR.DS-1, PR.DS-2, PR.DS-6, PR.PT-4, RS.MI-2 Thunderbird Mail User Agent Open (Mozilla) NLnet Labs PR.AC-1, PR.AC-5, PR.DS-1, PR.DS-2, PR.DS-6, PR.PT-4, RS.MI-2 Thunderbird with Apple Key Chain Secure64 Secure64 PR.AC-1, PR.AC-5, PR.DS-1, PR.DS-2, PR.DS-6, PR.PT-4, RS.MI-2 4.2.2 Email Servers Email servers include both Windows and Linux-based (Dovecot/Postfix) MTAs. Server-to-server encryption was demonstrated in the Postfix environments. Authentication of domain and server identity was based on DNSSEC-signed DANE records. Use of these DANE records is only supported by Postfix at the time of this project. The MTAs support each of the Cybersecurity Framework Functions, Categories, and Subcategories identified in section 3.4.4 above. The servers were demonstrated in different DNS environments and different TLSA RR usage scenarios. To demonstrate representative TLSA parameters, the demonstrations used self-signed certificates, end-entity certificates generated by well-known CAs and end-entities generated by enterprise local CAs. NIST SP 1800-6B: Domain Name System-Based Electronic Mail Security 44 This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1800-6. Table 4.2 Mail Transfer Agents Application Source Collaborator Configuration Support Cybersecurity Framework Category Exchange 201636 Mail Transfer Agent TLS Capable Microsoft Microsoft PR.AC-1, PR.AC-5, PR.DS-1, PR.DS-2, PR.DS-6, PR.PT-4, PR.CM-1, DE.CM-6, DE.DP-4, DE.RP-1, RS.CO-2, RS.MI-1, RS-MI-2 Postfix Mail Transfer Agent TLS Capable DANE Capable Open (postfix.com) NLnet Labs Fraunhofer Secure64 PR.AC-1, PR.AC-5, PR.DS-1, PR.DS-2, PR.DS-6, PR.PT-4, PR.CM-1, DE.CM-6, DE.DP-4, DE.RP-1, RS.CO-2, RS.MI-1, RS-MI-2 4.2.3 DNS Servers Both Windows and Linux-based DNS server and support components were contributed. DNS services provided include DNSSEC validating DNS resolvers (stub and recursive) and authoritative DNS servers for DNSSEC signed zones. Support for SMIMEA and TLSA records was demonstrated. The DNS server components support each of the Cybersecurity Framework Functions, Categories, and Subcategories identified in section 3.4.4 above with the exception of PR.DS-1 (protection of data-at-rest). Application Source Collaborator Configuration Support Cybersecurity Framework Category Active Directory and Windows Server 2016 ν Supports DNSSEC Microsoft Microsoft PR.AC-1, PR.AC-5, PR.DS-1, PR.DS-2, PR.DS-6, PR.PT-4, PR.CM-1, DE.CM-6, DE.DP-4, DE.RP-1, RS.CO-2, RS.MI-1, RS-MI-2 36 Exchange provided integrity protection only for PR.DS-1, PR.DS-2, and PR.PT-4 (Scenario 2). NIST SP 1800-6B: Domain Name System-Based Electronic Mail Security 45 This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1800-6. Application Source Collaborator Configuration Support Cybersecurity Framework Category BIND ν Supports DNSSEC ν Supports DANE Open (ISC) Internet Systems Consortium (ISC) PR.AC-1, PR.AC-5, PR.DS-1, PR.DS-2, PR.DS-6, PR.PT-4, PR.CM-1, DE.CM-6, DE.DP-4, DE.RP-1, RS.CO-2, RS.MI-1, RS-MI-2 NSD4 ν Supports DNSSEC ν Supports DANE Unbound ν Supports DNSSEC OpenDNSSEC Open (NLnet Labs) Open (NLnet Labs) PR.AC-1, PR.AC-5, PR.DS-1, PR.DS-2, PR.DS-6, PR.PT-4, PR.CM-1, DE.CM-6, DE.DP-4, DE.RP-1, RS.CO-2, RS.MI-1, RS-MI-2 DNS AUTHORITY DNS MANAGER ν Supports DNSSEC ν Supports DANE (Caching authority is labeled DNS CACHE, and signer runs on a dedicated processor) Secure64 Secure64 PR.AC-1, PR.AC-5, PR.DS-1, PR.DS-2, PR.DS-6, PR.PT-4, PR.CM-1, DE.CM-6, DE.DP-4, DE.RP-1, RS.CO-2, RS.MI-1, RS-MI-2 5 Outcome This section discusses the security platform from the perspective of the user and the system administrator. We define system administrator as a person within the organization who has elevated privileges on the management systems in the build. System administration functions include identification of system components, system installation, system integration, system configuration, configuration monitoring, identification of exception conditions, system maintenance, and status reporting to management. NIST SP 1800-6B: Domain Name System-Based Electronic Mail Security 46 This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1800-6. 5.1 The User’s Experience The user’s experience varies from relatively minimal additional impact in enterprise environments with established system administration and support to a significant impact in the case of individual selfsupported users. Where the enterprise offers systems administration and support services, the user’s experience with respect to DNS services is essentially unchanged. One exception is that, where DNSSEC authentication fails, email messages sent to or by a user will not be delivered. This should be an uncommon experience for correspondents but it is up to the enterprise DNS administrator to prevent this happening. Errors due to DNSSEC validation failures are not sent back to the end user and may not be logged at the sending MTA, but at the validating recursive resolver which detected the error. Similarly, for server-to-server encryption, the security protection features should be essentially transparent to the user. For user-to-user digital signature, the user must first have a certificate installed in their MUA. This may be included in digital identity credentials, or it may be provided by the system administrator in the process of provisioning the user’s computer. Otherwise, the procedure required would be similar to that followed in section 3.2 of SP 1800-6C. The steps required vary from platform to platform (e.g., Windows, Linux, Mac), user agent to user agent (e.g., Outlook vs Thunderbird) and how the private key is stored (on the system, smart cards, etc.) Representative user requirements are described below (in this case for Outlook running on MacBook and Thunderbird running on Linux.) 5.1.1 User’s Digital Signature Experience with Outlook on MacBook To use digital signatures and encryption, both the sender and recipient must have a mail application that supports the S/MIME standard (e.g., Outlook). Note: Before this procedure is started, a certificate must be added to the keychain on the computer. For information about how to request a digital certificate from a certification authority, see macOS Help or click on “Help” on the Outlook tool bar. 1. On the Tools menu, click Accounts. 2. Click the account that is to be used to send a digitally signed message, click Advanced, and then click the Security tab. 3. Under Digital signing, on the Certificate pop-up menu, click the certificate that is to be used. Note: The Certificate pop-up menu only displays certificates that are valid for digital signing or encryption that have already been added to the keychain for the macOS user account. To learn more about how to add certificates to a keychain, see macOS Help. 4. Do any of the following: NIST SP 1800-6B: Domain Name System-Based Electronic Mail Security 47 This publication is available free of charge from: https://doi.org/10.6028/NIST.SP.1800-6. To Do this Make sure that the digitally signed messages can be opened by all recipients, even if they do not have an S/MIME mail application and cannot verify the certificate Select the Send digitally signed messages as clear text check box. Allow the recipients to send encrypted messages to you Make sure that signing and encryption certificates have been selected on this screen, and then select the Include my certificates in signed messages check box. 5. Click OK, and then close the Accounts dialog box. 6. In an email message, on the Options tab, click Security, and then click Digitally Sign Message. 7. Finish composing the message, and then click Send. 5.1.2 User’s Digital Signature Experience with Thunderbird For purposes of illustration, the description of the user experience with Thunderbird also included certificate management requirements. The example here shows both S/MIME and PGP examples of certificate management. The S/MIME approach is recommended. Note that when using OpenPGP, a FIPS 140-conformant version should always be used.