Helmut Reiser*, Daniel Feuchtinger, Wolfgang Hommel, Bernhard Schmidt und Michael Storz

E-Mail made in Science — Sicherheit für den E-Mail Transport mit DANE TLSA

DOI 10.1515/pik-2015-0004

Abstract: Kein anderer Ansatz hat die Internet-basierte private und geschäftliche Korrespondenz so nachhaltig geprägt wie der E-Mail-Versand auf Basis des Protokolls SMTP. Die weltweite Mailserver-Infrastruktur kämpft allerdings seit Jahrzehnten mit Missbrauch, z. B. durch Spam und Phishing, und muss sich angesichts anhaltend nur sporadisch umgesetzter Ende-zu-Ende-Verschlüsselung spätestens seit den Snowden-Enthüllungen mit dem zuverlässigen vertraulichen Transport der ihr anvertrauten Daten auseinandersetzen. In diesem Artikel wird zunächst auf die betriebliche Notwendigkeit verschlüsselnden E-Mail-Transports u. a. aufgrund gesetzlicher Vorgaben und den aktuellen Stand der Technik mit seinen teils nicht immer offensichtlichen Defiziten eingegangen. Anschließend wird gezeigt, welche qualitativen Neuerungen sich auf Basis von DNSSEC mit dem Protokoll DANE TLSA ergeben. Am Beispiel der vom Leibniz-Rechenzentrum betriebenen Mailserver wird der praktische Einsatz von DANE TLSA diskutiert.

Einleitung

Schon lange bevor Telekommunikationsunternehmen damit begonnen haben die klassischen Telefonnetze konsequent auf IP-Telefonie umzurüsten, haben über das Internet verschickte E-Mails ihren Siegeszug angetreten. Die Ersparnis für dedizierte Geräte, die fast vernachlässigbar geringen Versandkosten, die i.d.R. sehr schnelle Zustellung und das einfache Erstellen von (Blind-)Kopien sind nur einige der Vorzüge gegenüber der ebenfalls asynchronen Kommunikation über Fax oder Post. E-Mails sind faktisch aus der privaten und geschäftlichen Korrespondenz nicht mehr wegzudenken und nehmen ihren festen Platz neben Kurznachrichtendiensten wie SMS und Instant-Messaging-Anwendungen sowie der (semi-)öffentlichen Gruppenkorrespondenz über soziale Netzwerke ein.

Wie beim Versand herkömmlicher Post müssen Absender und Empfänger von E-Mails aber der Transportinfrastruktur vertrauen. Der oft bemühte Vergleich von E-Mails mit Postkarten verdeutlicht, dass in Analogie zu Geräten und Personal der Post jeder, der am Transport einer E-Mail beteiligt ist, nicht nur deren zur Weiterverarbeitung benötigte Metadaten (wie Empfängeradresse und Umfang einer Sendung), sondern auch deren Inhalte einsehen könnte. E-Mail-Provider, die eine automatisierte Inhaltsanalyse z. B. zur Maßschneiderung von Werbeeinblendungen vornehmen, und bekannt gewordene Informationen über die Aufgabenbereiche und Arbeitsweisen u. a. von Nachrichtendiensten zeigen, dass das massenhafte Durchschnüffeln von E-Mails systematisch erfolgt und nicht auf Einzelfälle voyeuristischer Individuen beschränkt ist, denen bei Bekanntwerden ihres Handelns empfindliche Strafen drohen.

Die Verschlüsselung von E-Mail-Inhalten kommt, vordergründig mit nicht unberechtigter Kritik an der fehlenden intuitiven Nutzbarkeit entsprechender Software belegt, nur langsam in Gang; sie löst auch die Metadatenproblematik nicht, da passive Angreifer, die z. B. den gesamten Netzverkehr abhören, weiterhin analysieren können, wer mit wem wann von wo aus und in welchem Umfang per E-Mail kommuniziert. Die Betreiber von E-Mail-Servern kämpfen deshalb nicht mehr nur gegen aktiven Missbrauch der globalen E-Mail-Infrastruktur, z. B. durch Spam, Phishing und Malware-verseuchte E-Mails, sondern zunehmend auch gegen die passiven Bedrohungen, die sich aus der Überwachung sämtlicher Internet-Kommunikationsvorgänge ergeben.

Besonders bei Anti-Spam-Maßnahmen zeigt sich eine hohe Affinität der E-Mail-Infrastruktur zum Protokoll DNS. Bereits für seine grundlegende Funktionsweise benötigt SMTP dedizierte Informationen aus dem DNS in Form von Mail-Exchange-Resource-Records (MX-RRs), welche die für eine Domain zuständigen Empfangs-E-Mail-Server festlegen. Viele zur Spam-Abwehr eingesetzte Verfahren sind ebenfalls DNS-basiert: Zahlreiche Blacklists werden per DNS abgefragt und das Sender Policy Framework (SPF, RFC 7208) verwendet ebenso wie Domain-Keys Identified Mail (DKIM, RFC 6376) und Domain-based Message Au-

***Kontaktperson: Helmut Reiser:** E-Mail: reiser@lrz.de
Daniel Feuchtinger: E-Mail: daniel.feuchtinger@lrz.de
Wolfgang Hommel: E-Mail: wolfgang.hommel@lrz.de
Bernhard Schmidt: E-Mail: bernhard.schmidt@lrz.de
Michael Storz: E-Mail: michael.storz@lrz.de

thentication, Reporting, and Conformance (DMARC, RFC 7489) DNS Resource Records vom Typ TXT.

Während bei der Spam-Abwehr weitgehend davon ausgegangen werden kann, dass sich Angreifer vom Kaliber eines Geheimdienstes nicht in die Abläufe einmischen und die organisierte Spammer-Kriminalität keine Kontrolle über den DNS-Verkehr von Mailservern im Internet hat, ist die verschlüsselte Kommunikation mit und zwischen Mailservern kritischer zu betrachten. Hier zeigt sich, dass erst mit DNSSEC eine zuverlässige Basis errichtet wird, auf der neue Protokolle wie DANE TLSA aufsetzen können, die ihrerseits wiederum die Grundlage für eine vertrauliche Mailserver-Kommunikation mittels SMTP darstellen. In diesem Artikel wird zunächst zusammengefasst, welche Mindestanforderungen heutzutage an E-Mail-Transportverschlüsselung beispielsweise durch Datenschutzaufsichtsbehörden gestellt werden. Daran anknüpfend wird anhand des opportunistischen Sicherheitsmodells erläutert, welche signifikanten Defizite der durch Minimalanforderungen implizierte aktuelle Stand der Technik aufweist und warum herkömmliche Public-Key-Infrastructure-(PKI-)basierte Ansätze für TLS-gesicherte Verbindungen unzulänglich sind. Mit DANE TLSA (DNS-based Authentication of Named Entities TLS Authentication) wird ein DNSSEC-basierter Lösungsansatz vorgestellt, der authentifizierten, zwingend verschlüsselten E-Mail-Transfer ermöglicht, dabei verschiedenen Angriffsvarianten vorbeugt und auch für andere Protokolle wie HTTPS eingesetzt werden kann. Abschließend wird am Beispiel der vom Leibniz-Rechenzentrum betriebenen Mailserver gezeigt, dass in der Praxis bereits ein signifikanter Teil des Mailverkehrs über DANE TLSA abgesichert werden kann, obwohl DNSSEC bislang nur für einen Bruchteil aller Domains konfiguriert ist.

Anforderungen an verschlüsselten E-Mail-Transport

Welche Schutzmaßnahmen Organisationen beim Betrieb von Mailservern umsetzen, hängt oftmals von der persönlichen Einstellung der zuständigen Administratoren und den verfügbaren Ressourcen (Zeit, Budget, Know-How) ab. In der Praxis müssen Kompromisse aus den theoretisch aktuell möglichen, den aufgrund ihrer faktischen Verbreitung sinnvollen und den mit einem möglichst stabilen und effizienten Betrieb zu vereinbarenden Maßnahmen gefunden werden. Eine Objektivierung ist unter diesen organisationsvariablen Randbedingungen schwierig, so dass sich die weitere Diskussion pragmatisch daran orientiert, welche Anforderungen sich in jüngster Zeit mit Bezug auf gesetzliche Auflagen herauskristallisieren.

Das Bayerische Landesamt für Datenschutzaufsicht (BayLDA) hat im September 2014 eine Datenschutzüberprüfung der Mailserver (s. BayLDA-SMTP) von 2.236 Unternehmen mit Sitz in Bayern durchgeführt. Dabei wurde überprüft, ob die Mailserver

- verschlüsselte Übertragung von E-Mails (STARTTLS) anbieten,
- dabei Perfect Forward Secrecy (PFS) unterstützen und
- die Heartbleed-Lücke heartbleed gepatcht wurde.

Insgesamt 772 Unternehmen genügten den Anforderungen nicht und wurden vom BayLDA aufgefordert, die Missstände zu beseitigen. Das BayLDA begründet die Anforderungen damit, dass es Stand der Technik sei, E-Mails nur noch verschlüsselt zu übertragen.

Schauen wir uns daher die Anforderungen bzw. Testkriterien etwas genauer an. Zum Zeitpunkt der Überprüfung war die Heartbleed-Lücke bereits ein halbes Jahr alt. Bei einem funktionierenden Patch-Management sollte die Lücke daher seit langem geschlossen sein. In der Regel war dies wohl auch der Fall, da nur bei 44 Mailservern die Lücke noch vorhanden war. Eine zeitnahe Aktualisierung von Software mit Sicherheitslücken kann man sicherlich als Stand der Technik ansehen.

Damit eine E-Mail verschlüsselt übertragen wird, muss sowohl der Server (empfangender SMTP-Server) beim Aufbau einer Verbindung über die Eigenschaft STARTTLS (RFC 3207) die Unterstützung von Transport Layer Security (TLS) RFC 5246 dem Client (sendender SMTP-Server) anbieten als auch der Client dieses Angebot annehmen. Haben beide Parteien eine verschlüsselte Übertragung ausgehandelt, so sind sowohl die Metadaten wie z. B. Sender, Empfänger und Subject als auch der eigentliche Inhalt der E-Mail gegen passive Angriffe, also das Mitlesen der übertragenen Daten z. B. in einem Knotenpunkt des Netzes, geschützt.

Bei der automatisierten Onlineüberprüfung durch das BayLDA konnte prinzipbedingt nur das Verhalten der Server untersucht werden, da es über das Internet nicht möglich ist festzustellen, ob ein Client in der Lage wäre, über verschlüsselte Verbindungen zu senden. Offen bleibt aber die Frage, was passiert, wenn entweder die Server- oder Client-Software kein TLS unterstützt oder dieses nicht konfiguriert ist, während das Gegenüber dazu in der Lage wäre. Darf eine E-Mail dann unverschlüsselt verschickt werden? Wer verletzt dabei den Datenschutz, nur derjenige, der kein TLS kann, oder beide? Was ist, wenn einer der beiden Kommunikationspartner im Ausland sitzt? Beziehen sich die Anforderungen nur auf den Mailaustausch

zwischen zwei Mailservern (Message Transfer Agents = MTA) oder gelten sie auch für die Kommunikation eines beliebigen Mailclients (Mail User Agent = MUA) mit einem MTA, z. B. auf einen Drucker mit Scan-to-E-Mail-Funktion?

Der Einsatz von Perfect Forward Secrecy (PFS) verhindert, dass aufgezeichnete Daten im Nachhinein entschlüsselt werden können, wenn einer der privaten Langzeit-Schlüssel der Kommunikationspartner später kompromittiert wird. Auch hier ergeben sich analoge Fragen wie beim Einsatz von STARTTLS. Ist PFS wirklich Stand der Technik? So unterstützt z. B. die aktuellste im Juli 2015 veröffentlichte Version von MS Outlook für Mac OS X zwar modernere Chiffren scheint aber weiterhin noch kein PFS zu verwenden.

Zusammenfassend kann man also feststellen, dass das BayLDA als Stand der Technik den Einsatz von TLS nur gegen passive Angriffe sieht, wobei durch den Einsatz von PFS eine Massenüberwachung, z. B. durch Geheimdienste, erschwert werden soll. Zu Recht weist es aber darauf hin, dass es sich hierbei nur um Minimalanforderungen handelt, die keinen Schutz gegen aktive Man-in-the-middle-Angriffe (MITM) bieten. Ziel muss es aber sein, die Übertragung von E-Mails auch gegen solche Angriffe zu schützen. Für diesen Zweck werden im Folgenden Verfahren sowie der Stand der Technik vorgestellt.

Das opportunistische Sicherheitsmodell

In der Vergangenheit lag der Fokus bei der Entwicklung der Verschlüsselung von Internetprotokollen auf der maximalen Sicherheit einzelner Verbindungen. Es wurde daher ein Alles-oder-Nichts-Ansatz zum Schutz vor passiven wie auch aktiven Angriffen verfolgt. Damit war entweder vollständigen Schutz möglich oder eine Kommunikation war nicht bzw. nur unverschlüsselt möglich. Hatten einzelne Nutzer Probleme, zu kommunizieren, so führte das aber oft dazu, dass die Verschlüsselung vom Administrator für alle Nutzer ausgeschaltet wurde.

Seitdem Informationen zur Arbeit von Geheimdiensten wie der amerikanischen National Security Agency (NSA) vorliegen, zeigt sich aber, dass das Bedrohungsmodell anders gelagert ist. Es geht nicht mehr nur um den Schutz einzelner Verbindungen, sondern um den maximal möglichen Schutz aller Verbindungen innerhalb einer Protokollwelt, um dem flächendeckenden Angriff auf die Privatsphäre der Nutzer Einhalt gebieten zu können bzw. diesen soweit wie möglich zu erschweren RFC 7258.

Das opportunistische Sicherheitsmodell RFC 7435 geht davon aus, dass es auf dem Weg zu einer umfassenden sicheren Kommunikation mehrere Zwischenschritte gibt. Mit jedem einzelnen Kommunikationspartner wird dabei die höchste mögliche Sicherheitsstufe gewählt, die noch eine Kommunikation zulässt. Dadurch wird insgesamt ein höheres Gesamtsicherheitsniveau erreicht als bei der Alles-oder-Nichts-Lösung.

Die Sicherheitsstufen sind dabei durch die Kombination der Größen Verschlüsselung und Client- bzw. Server-Authentifizierung gekennzeichnet:

Stufe 1: Unverschlüsselt und nicht authentifiziert. Diese Kombination bietet keinerlei Schutz vor Angriffen: Angreifer können die gesamte Kommunikation mitlesen oder sogar ändern. Es ist auch unklar, mit wem man im Endeffekt kommuniziert.

Stufe 2: Verschlüsselt und nicht authentifiziert. Hiermit bekommt man Schutz vor passiven Angriffen, d. h. die Daten können nicht mehr abgehört werden; insbesondere eine Massenüberwachung ist nicht mehr so einfach möglich. Im Fall von TLS erhält man neben der Vertraulichkeit auch noch die Integrität der Daten, d. h. die Daten können auf dem Weg zum Empfänger nicht unbemerkt manipuliert werden.

Stufe 3: Verschlüsselt und authentifiziert. Mit Authentifizierung ist man neben passiven auch vor aktiven Man-in-the-Middle (MITM) Angriffen geschützt. Stufe 3 lässt sich wiederum in zwei Stufen unterteilen:

Stufe 3a: Nur der Server authentifiziert sich gegenüber dem Client (Einseitige Authentifizierung)

Stufe 3b: Client und Server authentifizieren sich gegenseitig (Zweiseitige Authentifizierung)

Abhängig vom jeweiligen Anwendungsfall wird als Basislinie eine der Stufen gewählt. Mindestens diese Stufe der Sicherheit muss erreicht werden, ansonsten kommt es nicht zu einer Kommunikation.

Im Mailbereich gibt es eine Reihe von Protokollen, mit denen E-Mails übertragen und durch TLS gesichert werden können:

- Zugriff auf den Messagestore (MS) durch einen MUA. Hier kommen die Protokolle IMAP(S) und POP3(S) (RFC 2595) zum Einsatz.
- Versand von E-Mails vom MUA an den Message Submission Agent (MSA). Hier wird eine Form von SMTP eingesetzt, die dienstmäßig als SUBMISSION(S) (RFC 4409) bezeichnet wird.
- Austausch von E-Mails zwischen MTAs mit Hilfe von SMTP.

Bei den Protokollen, bei denen ein MUA involviert ist, ist zur Nutzung des jeweiligen Dienstes immer eine Authentifizierung des Nutzers notwendig. Dafür könnte vom Mailprotokoll über den SASL External Mechanismus (Simple Authentication and Security Layer, RFC 4616) die Client-Authentifizierung von TLS genutzt werden. In der Regel wird dafür aber der Passwort-Mechanismus verwendet und nicht auf TLS zurückgegriffen. Da das Passwort nur verschlüsselt über die Leitung geschickt werden darf, kommt bei den MUA-Protokollen als Basislinie nur Stufe 2 in Frage.

Bei der MTA-zu-MTA-Kommunikation über das Internet muss als Basislinie weiterhin Stufe 1 genommen werden. Schaut man sich die Statistik in Googles Transparenzbericht google-transparenz an, so wurden im Mai 2015 54% der bei Google ankommenden und sogar 80% der abgehenden E-Mails verschlüsselt übertragen. Gegenüber den Daten vor einem Jahr (Juni 2014) hat sich die Rate bei den ankommenden verschlüsselt übertragenen E-Mails nicht geändert (54%), während die Rate bei den abgehenden E-Mails um gute 10% gestiegen ist. Diese Zahlen sind zwar sehr ermutigend und zeigen, dass der Weg in die richtige Richtung geht, aber sie reichen noch nicht aus, um allgemein die Basislinie auf Stufe 2 zu erhöhen.

Verschlüsselte Verbindungen

Beim Aufbau einer verschlüsselten Verbindung zum Transport von E-Mails muss zwischen dem Aufbau der TLS-Verbindung, die die Verschlüsselung bereitstellt, und der Interaktion zwischen dem TLS- und den Mailtransport-Protokollen unterschieden werden. Neben Angriffen auf den Aufbau der TLS-Verbindung, die nicht dienstspezifisch sind, interessiert vor allem, wie dedizierte Angriffe auf das Zusammenspiel der Protokolle abgewehrt werden können. Einen Überblick auf TLS-spezifische Angriffe findet sich in RFC 7457, eine Empfehlung wie TLS möglichst sicher eingesetzt werden kann in RFC 7525.

Aufbau einer verschlüsselten Mailtransport-Verbindung

Bei den meisten Mail-Protokollen gibt es zwei Möglichkeiten, um eine verschlüsselte Verbindung zu etablieren.

1 Der Server avisiert beim Verbindungsaufbau, dass er TLS unterstützt. Der Client akzeptiert das mit dem STARTTLS-Kommando und es wird eine TLS-Verbindung als Schutzhülle um das Mailprotokoll aufgebaut (Protokolle: IMAP, POP3, SUBMISSION, SMTP).

2 Bei den MUA-Protokollen wird über die Verwendung eines speziellen Ports signalisiert, dass zuerst eine TLS-Verbindung und anschließend erst die Mail-Verbindung innerhalb des Schutzes der TLS-Verbindung aufgebaut werden soll (die S-Varianten der Protokolle: IMAPS, POP3S, SUBMISSIONS). SMTP bietet standardmäßig diese Variante nicht an.

Im Folgenden wird erläutert, wie ein Angreifer den Aufbau einer verschlüsselten Verbindung vehindern kann.

STARTTLS-Downgrade-Attacke

Die erste Variante STARTTLS lässt sich durch einen aktiven Angriff auf recht einfache Weise aushebeln: Der Angreifer muss nur das Angebot des Servers von STARTTLS blockieren. Der Client baut daraufhin eine unverschlüsselte Verbindung auf. Firewalls, bei denen die Analyse des SMTP-Verkehrs (SMTP-Inspection) aktiviert ist (z. B., PIX/ASA von Cisco), filtern die STARTTLS Nachricht und schon kommt keine verschlüsselte Verbindung mehr in Gang.

Im Fall von SMTP, das bei der Kommunikation über das Internet zurzeit noch von der niedrigsten Basislinie (unverschlüsselte Verbindungen) ausgehen muss und ausschließlich die STARTTLS-Variante nutzt, gibt es momentan keine durch einen RFC unterstützte Abwehrmöglichkeit gegen diesen einfachen Angriff. Es lässt sich allenfalls im Nachhinein per Logfile-Analyse feststellen, dass eine Verbindung unverschlüsselt aufgebaut wurde. D.h. STARTTLS ohne sonstige Sicherheitsmechanismen kann für SMTP nicht als sicher angesehen werden.

Aber selbst wenn der Server auf Zwangsverschlüsseltung (Sicherheitsstufe 2) konfiguriert wurde, kann dies durch einen MITM-Angriff wie SSL-Stripping Marlinspike 2009 angegriffen werden. Hierbei wird ein Proxy eingeschaltet, der zum Server eine verschlüsselte Verbindung aufbaut, während er sich mit dem Client über eine unverschlüsselte Verbindung unterhält.

Es muss also auf beiden Seiten die Information über eine Zwangsverschlüsselung vorliegen, um sicher zu gehen, dass eine Downgrade-Attacke nicht möglich ist.

Abwehr einer STARTTLS-Downgrade-Attacke

Für eine Abwehr der STARTTLS-Downgrade-Attacke muss es – um im opportunistischen Sicherheitsmodell zu bleiben – möglich sein, pro Ziel-MTA bzw. Empfängerdomain festzustellen, ob zur Übertragung der E-Mail nur eine verschlüsselte Verbindung aufgebaut werden darf. Diese In-

formation liegt zum Zeitpunkt des Verbindungsaufbaus entweder bereits vor oder sie muss out-of-band über einen zweiten vertrauenswürdigen Kanal abrufbar sein.

Kommunizieren die Mailserver nicht über das Internet, sondern z. B. über ein Intranet, so kann man die Zwangsverschlüsselung fest in die Konfigurtion aufnehmen. Im Münchner Wissenschaftsnetz werden von den Border-MTAs des Leibniz-Rechenzentrums ca. 140 lokale Mailserver vor Spam und Viren aus dem Internet geschützt. Da der Relay-Dienst für jede einzelne Domain beantragt werden muss, kann somit auch die Information über die Zwangsverschlüsselung gleich mit in die Konfiguration der Mailserver aufgenommen werden.

Da es inband zum Zeitpunkt des Verbindungsaufbaus nicht möglich ist, an die Information zur Zwangsverschlüsselung zu kommen, kann dies inband nur über einen zeitlichen Versatz realisiert werden. Bei einer TOFU-Methode (trust on first use) geht man davon aus, dass beim erstmaligen Aufbau der Verbindung kein Angreifer aktiv ist. Man speichert die Information, immer eine verschlüsselte Verbindung aufzubauen, lokal ab und verwendet sie bei einem späteren Verbindungsaufbau. Versucht ein Angreifer zu diesem Zeitpunkt dann eine Downgrade-Attacke, so wird dies bemerkt und die Verbindung kommt nicht zustande.

Im Webbereich ist mit HTTP Strict Transport Security (HSTS) RFC 6797 eine entsprechende Technik definiert, bei der der Server dem Client mitteilt, dass er in Zukunft nur noch verschlüsselte Verbindungen zu ihm aufbauen soll.

Für die Mailprotokolle gibt es leider keinen entsprechenden Standard. Der Client kann zwar diese Information abspeichern und verwenden, muss aber damit rechnen, dass der Server aus welchen Gründen auch immer zwischendurch die Verschlüsselung abschaltet. Es ist dann ein manueller Eingriff notwendig, um das Problem mit der Gegenseite zu klären oder die Zwangsverschlüsselung abzuschalten.

Der nach unserer Kenntnis erste ISP, der die TOFU-Technik für den Versand von E-Mails im Produktionsbetrieb trotzdem einsetzt, ist der Mailbox Provider mailbox.org der Heinlein Support GmbH (s. mailbox.org). Er scheut den damit verbundenen höheren administrativen Aufwand, insbesondere in der Überwachung der Verbindungen, nicht, um höchstmögliche Sicherheit für seine Kunden zu erreichen und damit auch den Anforderungen an den Datenschutz zu genügen.

Eine andere Möglichkeit für das a priori Wissen über die Zwangsverschlüsselung ist der Einsatz der S-Varianten bei den Mailprotokollen. Durch die Verwendung des entsprechenden Ports ist von vornherein die Verwendung von TLS festgelegt. Ein Fallback auf die STARTTLS-Variante des Protokolls ist nicht vorgesehen und daher auch kein Downgrade möglich. Leider funktioniert diese Möglichkeit für SMTP mangels eines per RFC standardisierten Ports nicht.

In einem Intranet ist das hingegen möglich. So wurde bei DE-Mail nicht auf SMTP mit STARTTLS gesetzt, sondern man definierte einen SMTPS-Dienst auf Port 1465 in Anlehnung an SUBMISSIONS (Port 465). Zu einer Domain findet man den zugehörigen Mailserver über einen SRV-Record mit dem Label smtp (wobei dieses Label korrekterweise eigentlich smtps heißen müsste) (BSI TR 01201 Teil 1.4).

Ein zweiter Kanal kann z. B. ein proprietäres Protokoll sein, auf das sich eine Gruppe von Mailserver-Betreibern geeinigt hat. Als Beispiel sei hier der Zusammenschluss der Betreiber bei EmiG (E-Mail made in Germany) genannt. Die daran beteiligten Provider haben sich auf ein Vorgehens- und Teilnahmemodell geeinigt, dass aber leider weder für andere Maildienstleister offen zu sein scheint, noch wirklich transparent oder standard-basiert ist. Auch wenn ein großer Teil der Mail-Kommunikation innerhalb von Deutschland zwischen diesen Providern abgewickelt wird, skaliert dieser Ansatz für die weltweite Anwendung nicht. Hierzu bedarf es öffentlicher Standards, die auch jeder Betreiber eines Mailservers nutzen kann.

In der IETF Working Group DANE DANE-WG ist mit der Entwicklung von „SMTP security via opportunistic DANE TLS“ ein solcher Standard in Sicht. Bei DANE ist es möglich, über eine per DNSSEC abgesicherte DNS-Anfrage festzustellen, ob zu einem Mailserver immer eine TLS-Verbindung aufgebaut werden soll (s.u.).

Verhinderung von MITM Angriffen: Authentifizierte TLS-Verbindungen

Um wirklich gegen aktive MITM-Angriffe resistente verschlüsselte Verbindungen für die Übertragung einer E-Mail zu bekommen, müssen sich die Kommunikationspartner gegenseitig authentifizieren. Beim Aufbau einer TLS-Verbindung wird daher im Normalfall immer versucht, den Server zu authentifizieren. Die Authentifizierung des Clients hingegen ist optional und kommt im Mailbereich bisher nur selten zur Anwendung.

Ist der Client ein MUA, so wird hierfür wie oben erläutert in der Regel die Passwort-basierte Authentifizierung innerhalb des Mailprotokolls genutzt und die Client-Authentifizierung von TLS nicht benötigt.

Auch bei der Übertragung einer E-Mail im Internet von einem MTA zum nächsten kommt die Authentifizierung des Clients zurzeit nicht zur Anwendung. Das liegt daran,

dass eine E-Mail auf dem Weg zur ihrem Ziel über mehr als einen MTA geleitet werden kann, z. B. bei Mailinglisten oder Weiterleitungen. Durch diesen indirekten Mailfluss wird die Zuordnung von Absenderadresse zu sendendem Client-MTA aufgebrochen, sodass der Server-MTA nicht mehr überprüfen kann, ob der Client-MTA berechtigt ist, E-Mails mit dieser Absenderadresse zu senden. Deswegen wird in der Regel auf die Client-Authentifizierung verzichtet. Mit dem Protokoll DMARC bzw. den darunterliegenden Protokollen SPF und DKIM wird versucht, diese Überprüfung durchzuführen. Wie man an der seit ca. 2 Jahren im Kreise verlaufenden Diskussion zum Update des DMARC-Protokolls feststellen kann, kommt aber auch DMARC mit dem indirekten Mailfluss nicht zurecht.

Die Authentifizierung des Client-MTAs wird allenfalls dazu genutzt, um ihm eine Relay-Erlaubnis über den Server-MTA zu erteilen. So können die lokalen Mailserver im Münchner Wissenschaftsnetz die Mailrelays des LRZ zum Versand ihrer E-Mails nutzen. Natürlich müssen die Mailrelays wissen, welche MTAs diesen Dienst nutzen dürfen, ansonsten könnte jeder Spammer dies für seine Zwecke ausnutzen. Aber auch hier wird in der Regel für die Autorisation die IP-Adresse des Client-MTAs statt des Domainnamens genutzt.

Beim Aufbau einer TLS-Verbindung authentisiert sich der Server mittels eines X.509-Zertifikates, dass vom Client validiert werden muss. Das alleine reicht aber nicht aus, der Client muss auch sicher überprüfen können, ob der Server wirklich für Mails der Domain zuständig ist. Ist die Überprüfung erfolgreich, wird ein Server-seitiger MITM-Angriff, in gewissem Rahmen, verhindert.

Probleme bei Public Key Infrastruktur mit X.509-Zertifikaten (PKIX) und Mail

Das Management der Zertifikate wird durch eine Public Key Infrastruktur (PKIX) ermöglicht. Sie regelt die Ausgabe und die Sperrung von Zertifikaten und ermöglicht die Überprüfung der Zertifikate auf Gültigkeit RFC 5280.

End Entity (EE-) Zertifikate binden ein „Subjekt“ an einen öffentlichen Schlüssel. Das Subjekt ist die Entität, die den privaten Schlüssel besitzt. Zertifikate werden von einer Zertifizierungsstelle (Certificate Authority, CA) ausgestellt. Sie überprüft bei der Antragstellung, ob das Subjekt auch wirklich zu dem privaten Schlüssel gehört und signiert zur Bestätigung das Zertifikat mit ihrem privaten Schlüssel. Ihr Zertifikat mit ihrem öffentlichen Schlüssel signiert sie entweder selbst – es wird damit zu einem Root-Zertifikat – oder es wird wiederum von einer anderen CA ausgegeben und signiert, d. h. es entsteht eine Kette von Zertifikaten vom EE-Zertifikat bis zum Zertifikat der Root-CA. Um ein Zertifikat zu validieren, muss die ganze Zertifikatskette überprüft und dem Zertifikat der Root-CA (= Vertrauensanker, engl. Trust Anchor) vertraut werden.

Im Web hat das CAB-Forum Richtlinien und Anforderungen an Zertifikate entwickelt, die von CAs eingehalten werden müssen, damit ihre Root-Zertifikate in den jeweiligen Browser integriert werden. Einige große Browser- und andere Hersteller, z. B. Adobe, Apple, Google, Microsoft, Mozilla und Oracle sind Mitglieder des CAB-Forums und spezifizieren Akzeptanzverfahren, mit denen sich CAs um die Aufnahme ihrer Root-Zertifikate in den entsprechenden Trust Anchor Store (Microsoft: Trusted Root Certification Authorities Store) bewerben können. Diese TA-Stores werden wiederum von anderen Herstellern von Browsern oder Betriebssystemen genutzt. So nutzen Debian- und SuSE-Linux im Wesentlichen die Zertifikate aus dem TA-Store von Mozilla. Damit bestimmen im Grunde die im CAB-Forum vertretenen Firmen, welche Vertrauensanker und damit welche Zertifikate als vertrauenswürdig anzusehen sind.

In der Praxis ist die Konstruktion der möglichen Zertifikatsketten zwischen der Root-CA und einem EE-Zertifikat und die Überprüfung dieser Ketten wegen der zahlreich möglichen Nebenbedingungen, wie Name- oder Policy-Constraints, ein hoch komplexer Vorgang, bei dem viele Fehler gemacht werden können und in der Vergangenheit auch gemacht wurden. Diese Fehler führen dann dazu, dass einem Zertifikat fälschlicherweise vertraut wird.

Sperrung von Zertifikaten

Ein weiteres Problem für die alleinige Verwendung von PKIX im Mail-Bereich ist der Widerrruf von Zerfitikaten.

Es gibt eine Reihe von Gründen, warum ein Zertifikat bereits vor Ablauf seines Gültigkeitsdatums aus dem Verkehr gezogen werden soll. Der wichtigste ist sicher die Kompromittierung des privaten Schlüssels. Einer der schwerwiegendsten Fälle war im April 2014 die Aufdeckung des Heartbleed-Fehlers in OpenSSL. Allein in der ersten Woche, nachdem der Fehler bekannt wurde, wurden zur Vorsicht mehr als 80.000 Web-Server-Zertifikate gesperrt (Netcraft-Heartbleed).

Zur Überprüfung, ob ein Zertifikat gesperrt wurde, gibt es eine Reihe von Möglichkeiten:

- Certificate Revocation List (CRL) RFC 5280,
- Online Certificate Status Protocol (OCSP) RFC 6960,
- OCSP stapeling, RFC 6066 (Version 1) bzw. RFC 6961 (Version 2),
- Proprietäres Protokoll für eine Sperrliste.

In einer CRL hält eine CA fest, welche Zertifikate gesperrt sind. Zur Überprüfung muss die CRL zuerst aus dem Internet geladen werden, sofern nicht eine Kopie im Cache bereits vorliegt. Das CAB-Forum spezifiziert als maximale Cachezeit 10 Tage für EE-Zertifikate und 12 Monate für Zwischen-CA-Zertifikate. Befindet sich also eine CRL im Cache, so kann es insbesondere bei einem kompromittierten CA-Zertifikat sehr lange dauern, bis sich die Sperre auswirkt.

Das zweite Problem ist die Größe der Sperrliste. Als CloudFlare infolge des Heartbleed-Fehlers alle Zertifikate für ihre Kunden neu ausstellte, wuchs die CRL ihrer primären CA GlobalSign von 22 KByte auf 4,7 MByte an. CloudFlare schätzt, dass durch die Größe der CRL zusätzliche monatliche Kosten in Höhe von 400.000$ für den erhöhten Internetverkehr beim Abruf der CRL verursacht wurden (CloudFlare-Cost).

Für die Überprüfung einer Sperrung in Echtzeit wurde OCSP entwickelt. Bei jedem Aufbau einer TLS-Verbindung muss für jedes Zertifikat aus der Kette eine Anfrage beim OCSP-Server der zuständigen CA gemacht werden. OCSP-Antworten können aber auch wieder in einem Cache abgelegt werden. Die Cachezeiten für OCSP entsprechen denen des CAB-Forums. Von einer Sperrung in Echtzeit kann also nicht die Rede sein, wenn eine CA diese Maximalzeiten nutzt. Zudem ergibt sich eine Reihe von Problemen:

- Datenschutz: Die CA weiß durch OCSP-Anfragen, wann zwischen welchen TLS-Partnern eine Verbindung aufgebaut wurde.
- Serverlast: Die CA weiß bei der Ausstellung eines Zertifikats nicht, welche Last auf ihren OCSP-Server zukommt. Kommt es zu Lastproblemen, werden Anfragen nicht mehr beantwortet. Alle Browser implementieren daher in solchen Fällen einen Soft- und keinen Hard-Fail.
- Ein Angreifer kann die Abfrage blockieren bzw. mit der Antwort *trylater* zurückschicken. Browser geben dann keine Fehler zurück, sondern bauen trotz fehlender Antwort die Verbindung auf.

Mit OCSP stapeling wird versucht die Nachteile von OCSP zu vermeiden. Hier wird mit dem Zertifikat gleichzeitig eine OCSP-Antwort mitgeschickt. In Version 1 ist dies nur für das EE-Zertifikat möglich, für die Zwischen-CA-Zertifikate muss weiterhin der OCSP-Server oder die CRL abgefragt werden. Mit Version 2 kann für alle Zertifikate eine OCSP-Antwort geschickt werden.

Problematisch ist dabei aber, dass ein Angreifer die mitgeschickte OCSP-Antwort unbemerkt ausfiltern kann. Daher versucht man zurzeit, eine neue TLS-Erweiterung inklusive Zertifikatserweiterung zu definieren, die das Ausfiltern verhindern soll (Must-Staple). Mit OCSP must-staple wird die Gültigkeitsdauer eines Zertifikats, die sich normalerweise in Jahren bemisst, in kleine, maximal 10 Tage gültige Zeiträume zerteilt. Ein Zertifikat muss dann nicht mehr gesperrt werden, sondern verliert einfach seine Gültigkeit.

Ein global verlässlicher und zuverlässiger Widerruf von Zertifikaten, der automatisiert verarbeitet werden könnte, ist derzeit nicht gegeben (zur Situtation bei HTTP, s. [CRLSet]). Dies stellt insbesondere bei der Kommunikation zwischen Mailservern ein großen Problem dar. Im Gegensatz zur Kommunikation im Web, bei der man ggf. über ein Pop-Up den Nutzer entscheiden lassen kann, ist im Mailbereich ein vollständig automatisierbarer Ansatz unerläßlich.

Kompromittierte Certification Authorities

Ein noch grundsätzliches Problem im CA-Modell besteht darin, dass weltweit jede CA ein Zertifikat für eine beliebige Domain ausgeben kann, d. h. es besteht keine Verzahnung mit der DNS-Struktur. Es besteht zwar theoretisch die Möglichkeit, CAs über Name Constraints auf Teilbäume im DNS zu beschränken. Dies kommt aber allenfalls auf der Ebene von untergeordneten CAs zur Anwendung. Daher ist das Gesamtsystem nur so sicher wie die „schwächste“ CA, deren Zertifikat man vertraut. In der Vergangenheit kam es bereits zu zahlreichen kompromittierten CAs. Einer der schwerwiegendsten Vorfälle war 2011 der Einbruch bei DigiNotar, bei dem ca. 500 falsche Zertifikate, u. a. für die Domains mail.google.com, login.yahoo.com, www.google.com, login.live.com, addons.mozilla.org, login.skype.com, erzeugt wurden. In anderen Fällen wurden untergeordnete CA-Zertifikate missbraucht, um Zertifikate für z. B. Google auszustellen, so etwa in 2013 bei TURKTRUST Inc. und bei der CA des Government of France (ANSSI).

Certificate Transparency

Nachdem die fälschlicherweise ausgestellten Zertifikate oft Domains von Google betrafen, holte Google 2013 mit RFC 6962 (befindet sich momentan in der Überarbeitung Draft CT, weitere Information auch unter www.certificate-transparency.org) zum Gegenschlag aus. Google schlägt vor, die Ausgabe sämtlicher Zertifikate der öffentlichen CAs zu überwachen. Die CAs oder die Eigentümer der Zertifikate sollen diese mit der gesamten Zertifikatskette an einen oder mehrere Log-Server übermitteln. Aber auch jeder andere Internetteilnehmer kann Zertifikate beim Log-Server melden. So registriert Google bereits jedes Zertifikat, das es bei der Indizierung des Internets findet, in

seinen selbst betriebenen Log-Server. Im Gegensatz zur Anzahl an CAs sollen weltweit nur eine Handvoll Log-Server existieren, die vermutlich oft von den Betreibern der TA-Stores betrieben werden.

Eigentümer von Domains können jederzeit über das Internet bei einem Log-Server nachfragen, ob zu einer ihrer Domains ein Zertifikat registriert wurde. Handelt es sich um ein unerlaubt ausgestelltes Zertifikat, so kann der Eigentümer von der ausstellenden CA verlangen, dass das Zertifikat für ungültig erklärt wird. Reagiert die CA nicht oder kommt es zu mehreren solcher Fälle, können die Betreiber der Log-Server und/oder der TA-Stores beschließen, die Vertrauensanker für diese CA aus ihrem TA-Store zu eliminieren (Log-Server haben ihren eigenen TA-Store, nur für die darin enthaltenen Vertrauensanker nehmen sie Zertifikate an). Damit können für diese CA beim Log-Server keine Zertifikate mehr registriert bzw. die Zertifikatsketten können nicht mehr validiert werden.

Was hindert einen Angreifer daran, sein falsches Zertifikat einfach nicht registrieren zu lassen? Als Druckmittel sollen die TLS-Clients fungieren. Bei der Registrierung des Zertifikats stellt der Log-Server eine Signed Certificate Timestamp (SCT) aus. Schickt ein TLS-Server beim Aufbau einer TLS-Verbindung sein Zertifikat, so muss er gleichzeitig auch den SCT von einem oder mehreren Logs mitschicken. Bekommt der TLS-Client keinen SCT geschickt oder ist dieser ungültig, so muss er den Aufbau der Verbindung ablehnen. Will der Angreifer sein Zertifikat nutzen, muss er es daher registrieren und gerät damit in die Gefahr, dass sein zu unrecht ausgestelltes Zertifikat entdeckt wird. Certificate Transparency verhindert somit nicht den Missbrauch eines fälschlich ausgestellten Zertifikats, sondern verkürzt die Zeit bis es entdeckt wird. Das eigentliche Ziel von Certificate Transparency ist aber die Identifikation von missbrauchten CA-Zertifikaten, um diese sperren bzw. aus den TA-Stores entfernen zu können.

Solange aber nicht ein großer Teil der TLS-Clients SCTs honorieren, sind die CAs bzw. Eigentümer der Zertifikate nicht gezwungen, an CT teilzunehmen. Certificate Transparency zielt vor allem auf den Webbereich. Dort ist die Überprüfung von Zertifikaten wesentlich weiter fortgeschritten als im Mailbereich. Nachdem Web-Clients es bisher zwar schwerer machen, eine TLS-Verbindung zu einem Server mit ungültigem Zertifikat aufzubauen, dies aber nicht verhindern, ist es unwahrscheinlich, dass sich das bei der Beachtung eines SCTs ändern wird. Es wird nur zu weiteren Fehlermeldungen kommen, die der Nutzer ignoriert. Bis Certificate Transparency wirkungsvoll gegen kompromittierte Zertifikate hilft, wird noch viel Zeit vergehen.

Certificate/Public Key Pinning

Bei der TOFU-Methode Certificate Pinning geht man noch einen Schritt weiter als bei der Abwehr des STARTTLS-Downgrade-Angriffs. Statt sich nur zu merken, dass die anderen Seite Verschlüsselung anbietet, merkt man sich, welches Zertifikat bzw. welcher öffentliche Schlüssel zum Einsatz gekommen ist. Das Zertifikat wird somit an den Domainnamen des Verbindungsendpunktes „angesteckt“, dies bezeichnet man als einen Pin. Wird bei einem späteren Verbindungsaufbau ein anderes Zertifikat vorgelegt, so wird der Verbindungsaufbau abgebrochen. Damit ist man vor einem MITM-Angriff geschützt, gleichzeitig ist es aber nicht mehr möglich, ein Zertifikat legal auszutauschen, z. B. weil die Gültigkeitsdauer abgelaufen ist oder der private Schlüssel kompromittiert wurde.

Da man in der Regel bei der Erneuerung eines Zertifikats das Schlüsselpaar beibehält – sofern es nicht kompromittiert wurde – empfiehlt es sich, bei EE-Zertifikaten nicht das ganze Zertifikat, sondern nur die Information zum öffentlichen Schlüssel (Subject Public Key Info (SPKI)) zu pinnen. Man ist dann gegen diese Art des Zertifikatsaustauschs immun. Beim Pinnen von CA-Zertifikaten, die seltener ausgetauscht werden, kann aber auch das ganze Zertifikat gepinnt werden.

Zusätzlich braucht man eine Lösung für den Austausch des öffentlichen Schlüssels. Dafür bekommt ein Pin eine Verfallszeit von ein paar Tagen, die beim erneuten Auftreten des Pins jedes Mal verlängert wird. Funktioniert das Pinning mit Unterstützung der Gegenseite, so kann diese bei einem geplanten Austausch des Schlüssels die Cachezeit heruntersetzen, analog zur Änderung eines MX-Records, bei der die TTL vorher ebenfalls auf einen kleinen Wert gesetzt werden sollte.

Certificate Pinning im Münchner Wissenschaftsnetz (MWN)

Im MWN wird für die Anbindung der ca. 140 Mailserver eine einfache Art des Pinnings verwendet (damit wird auch gleichzeitig die oben beschriebene Zwangsverschlüsselung realisiert). Da die meisten Mitglieder des DFN-Vereins auch an der DFN-PKI teilnehmen, betreiben sie jeweils eine eigene Unter-CA zur DFN-CA. Das LRZ pinnt nicht die EE-Zertifikate der jeweiligen Mailserver, sondern die jeweiligen CA-Zertifikate und nutzen daher keine Verfallszeit. Eine Änderung der CA-Zertifikate bekommen wir auf anderem Wege mit.

Diese Art des Pinnings könnte man noch auf alle Mailserver der Teilnehmer der DFN-PKI ausdehnen, sofern sicher gestellt ist, dass man über einen sicheren

Kanal von der Änderung der CA-Zertifikate informiert wird. Für eine allgemeine Anwendung für alle Mailserver im Internet ist diese einfache Art des Pinnings ohne Unterstützung durch den Server allerdings nicht ausreichend skalierbar.

Trust Assertions for Certificate Keys (TACK)

Soll eine Pinning-Methode für beliebige Applikationsprotokolle funktionieren, muss sie auf der Ebene des TLS-Protokolls arbeiten. Mit TACK gab es 2013 einen entsprechenden Vorschlag. Dabei wird nicht direkt auf das Zertifikat bzw. seine SPKI gepinnt, sondern auf ein vom Server-Administrator bestimmten TACK Signing Key (TSK). Mit dem TSK wird die SPKI signiert. In einem TACK werden der öffentliche Schlüssel des TSK, die Signatur und ein paar weitere Meta-Informationen gespeichert. Im TLS-Handshake schickt der Server ähnlich wie bei OCSP-stapeling mit dem Zertifikat auch den zugehörigen TACK, den der Client überprüfen kann.

Mit TACK ist es möglich einen Schlüssel auszutauschen, da der TSK gleich bleibt. Da der private Schlüssel des TSKs nur einmal bei der Signatur der SPKI benötigt wird und die restliche Zeit sicher in einem Tresor liegen kann, ist die Kompromittierung des TSK wesentlich unwahrscheinlicher als bei dem privaten Schlüssel, der bei jedem TLS-Verbindungsaufbaus benötigt wird. Mit der oben angesprochenen Meta-Information ist es aber auch möglich, den TSK geordnet auszutauschen.

Es gibt zwar einige Code-Beispiele, wie TACK in OpenSSL, Apache, nginx und Chromium eingebunden werden kann (s. http://tack.io/), aber leider hat der Vorschlag bisher nicht seinen Weg in eine Produktionsumgebung gefunden.

Überprüfung des Zertikats-Subjekts

Liegt ein gültiges Zertifikat vor, d. h. ist die Zertifikatskette in Ordnung und ist es weder abgelaufen noch gesperrt, so muss noch überprüft werden, ob das Subjekt des Zertifikats zu dem Server passt, zu dem die Verbindung aufgebaut werden soll. Dazu werden zwei Mengen gebildet.

1 Die Menge der „Presented Identifier“ wird aus allen Identifiern des Subjekts, also dem CN-ID und den Komponenten des Feldes *subjectAlternativeName* (SAN) gebildet. Die Elemente eines SANs sind typisiert. Der häufigste Typ (= Namensform) ist bei TLS der DNSName, der einen DNS-Domainname (DNS-ID) enthält.
2 Die Menge der „Reference Identifier“, d. h. der Identifier des Ziels.

Haben diese beiden Mengen eine nicht-leere Schnittmenge, dann ist sichergestellt, dass die Verbindung zum richtigen Server aufgebaut wurde und die TLS-Verbindung ist damit authentifiziert.

Die allgemeinen Regeln zu Bildung und Vergleich der Identifier sind in RFC 6125 beschrieben, während die Nutzung der SRV-Records und der SRV-IDs für die MUA-Protokolle im RFC 6186 und im Draft DANE-SRV enthalten sind.

Server Name Indication (SNI)

Um die Menge der Presented-Identifier zu bestimmen, wird das Zertifikat des Servers benötigt. Beim Verbindungsaufbau schickt der Server normalerweise sein mit der IP-Adresse verbundenes (Default-)Zertifikat zurück. Auf einem Server können aber gleichzeitig viele verschiedene virtuelle Server laufen, jeder mit einem eigenen Domainnamen. Gehören alle Domains zur selben Organisation, so können alle Domainnamen ohne Probleme im Feld *subjectAlternativeName* des Zertifikats aufgenommen werden. Das LRZ nutzt, z. B. im Webbereich, Zertifikate mit mehreren hundert Domainnamen.

Wird der Server aber bei einem ISP gehostet, so gehören die Domains in der Regel zu verschiedenen Organisationen. Der ISP müsste dann im Auftrag aller seiner Kunden bei einer CA das Zertifikat beantragen, damit alle Domainnamen zusammen in ein Zertifikat aufgenommen werden können. Sowohl aus Sicherheitsaspekten als auch organisatorisch ist es besser, wenn zu jedem virtuellen Kundenserver ein eigenes Zertifikat – und damit auch ein eigenes Schlüsselpaar – existiert. Damit der Server sich mit dem richtigen Zertifikat ausweisen kann, muss der Client dann aber dem Server signalisieren, welche Domain er erreichen will. Diese Funktion heißt Server Name Indication (SNI) und wird über eine TLS-Erweiterung vom Type *server_name* realisiert. (RFC 6066).

Dienst-spezifische Zertifikate

Lagert man einen Dienst an einen ISP aus, so will man oft, dass der ISP das zugehörige Zertifikat nur für diesen Dienst einsetzen kann. Die im Zertifikat vorhandenen Felder *key usage* und *extended key usage* können dafür nicht verwendet werden, da ersteres die kryptographischen Eigenschaften des Schlüssels beschreibt, z. B. die Eigenschaft “kann zum Verschlüsseln und/oder Signieren genutzt werden”, und zweiteres, in welchem Bereich das Zertifikat verwendet werden kann. Für TLS sind hier

der Einsatz zur Client- bzw. Server-Authentifizierung spezifiziert.

In RFC 4985 wurde daher zusätzlich eine weitere Namensform *SRVName* definiert, bei der dem Domainnamen noch der Dienst als Subdomain vorangestellt wird. Diese Identifier werden als SRV-IDs bezeichnet. So wird z. B. der Dienst IMAPS für die Domain example.com als _imaps.example.com angegeben. Das Zeichen „_“ dient zur Unterscheidung zu einem Server mit dem Domainnamen imaps.example.com.

Bestimmung der Reference Identifier

Reference-Identifier bestehen aus einem Domainnamen und in manchen Fällen zusätzlich aus einem Applikationstyp als Kennzeichnung für den jeweiligen Dienst. Ausgangspunkt bei der Bestimmung der Liste an Reference-Identifiern ist die Source-, Service- bzw. (original) Next-Hop-Domain (Bezeichnung aus RFC 6125, DANE-SRV bzw. DANE-SMTP). Aus dieser wird die Derived Domain, Target Server Host Domain (TSHD) bzw. MX Hostname, also der Domainname des Zielservers, bestimmt. Bei den MUA-Protokollen besteht die Service-Domain aus der Domain der Absende-, bei SMTP aus der Domain der Empfangsadresse.

Wird bei den MUA-Protokollen der Domainname des Mailservers direkt vom Nutzer im Mail-Client konfiguriert oder bei SMTP alle E-Mails oder E-Mails an eine bestimmte Empfängerdomain über einen in der Konfiguration festgelegten Relay-MTA geschickt, so ist die Zuordnung der Service- zur target server host domain fest auf dem Client definiert. Je nach Implementation besteht die Liste der Reference-Identifier aus der Service- und der target server host domain oder nur aus der TSHD.

Wird die Bestimmung der TSHD hingegen über eine oder mehrere DNS-Abfragen realisiert, so haben wir eine initiale Domain und eine abgeleitete Domain. Die initiale Domain wird immer als Reference-Identifier verwendet. Ob die abgeleitete Domain zusätzlich hinzu kommt, hängt davon ab, ob jeder einzelne DNS-Record auf dem Weg von der initialen zur abgeleiteten Domain über DNSSEC abgesichert ist. Nur wenn dies der Fall ist, kann die abgeleitete Domain ebenfalls genutzt werden.

Ist hingegen einer der Records nicht abgesichert, so könnte ein Angreifer über DNS-Spoofing falsche Information unterjubeln. Daher darf bei unsicherem Ergebnis die abgeleitete Domain nur zum Aufbau der TLS-Verbindung, nicht aber für die Bestimmung der Referenz-Identifier verwendet werden.

DNS-Abfragen sind notwendig,

- wenn die Service-Domain oder die TSHD ein Alias (CNAME) sind,
- wenn bei den MUA-Protokollen der für eine Service-Domain zuständige Server über einen SRV-Record bestimmt , oder
- wenn bei SMTP der nächste Hop über einen MX-Record ermittelt wird.

Obwohl ein CNAME für die TSHD eines MX- oder SRV-Records nicht erlaubt ist, werden in der Praxis meistens auch CNAMEs verarbeitet. Damit kann die Liste der Reference Identifier theoretisch aus den vier Elementen

- initiale und expandierte Service-Domain
- initiale und expandierte TSHD

bestehen. War ein SRV-Record mit im Spiel, so können zusätzlich noch

- initiale und expandierte Service-Domain mit Appikationstyp

hinzukommen, wie in RFC 6125 vorgeschlagen wird. Leider ist weder in RFC 6186 noch in DANE-SRV genau beschrieben, welche der verschiedenen Service-Domain-Formen genutzt werden sollen, sodass es zu inkompatiblen Implementierungen kommen kann.

Ähnlich sieht es bei einem MX-Record aus. Man kann zwar aus DANE-SMTP gewisse Folgerungen zu den zu nutzenden Reference-Identifiern ziehen, genau ist das aber nicht festgelegt, da sich die Drafts zu DANE in erster Linie auf das neue DANE-Protokoll beziehen (s.u.). Meistens wird nur die initiale TSHD genutzt, so dass im ungesicherten Fall immer ein MITM-Angriff möglich ist. Selbst wenn im ungesicherten Fall des MX-Records immer die Service-Domain genutzt würde, müsste bei einer Auslagerung an einen ISP sowohl von der Client- wie von der Serverseite SNI unterstützt werden, was zum größten Teil nicht der Fall ist. Zusätzlich existiert das Problem, dass es keine dienstspezifischen Reference-Identifier gibt, obwohl ein MX-Record eigendlich nur ein spezieller SRV-Record ist.

Beim Vergleich der Reference- mit den Presented-Identifiern aus dem Subjekt des Zertifikats werden die Domains mit den Domains aus den DNS-IDs und die Reference-Identifier mit Applikationstyp mit den SRV-IDs verglichen.

Nehmen wir an, ein über IMAPS zugreifbarer Mailserver mit Namen mail.example.net ist für Mailadressen der Form user@example.com zuständig und lässt sich über die Abfrage eines SRV-Records finden. Dann wird die Menge der Reference-Identifier bei ungesichertem Record aus den beiden Domains imaps.example.com und example.com gebildet. Das Zertifikat sollte mindestens einen der

folgenden Identifier für einen Match enthalten: SRV-ID imaps.example.com, DNS-ID example.com und evtl. aus Kompatibilitätsgründen mit älteren Implementation die CN-ID example.com. Im gesicherten Fall kommt zu den Reference Identifiern noch die Domain mail.example.net hinzu und im Zertifikat die DNS-ID mail.example.net und evtl. die CN-ID mail.example.net.

Fazit zur zertifikatsbasierten Authentifizierung

Zusammenfassend kann man zum Stand der Technik sagen, dass die Situation bei den MUA-Protokollen etwas besser aussieht als bei SMTP. Für die MUA-Protokolle ist durch die Wahl des Ports möglich, die Zwangsverschlüsselung zu signalisieren und es existieren für die indirekte Bestimmung der Target Server Host Domain Vorgaben (obwohl nicht genau genug), welche Domainnamen in den Zertifikaten für eine Authentifizierung vorhanden sein sollen.

Bei SMTP hingegen gibt es weder die Möglichkeit, eine Zwangsverschlüsselung zu signalisieren, noch eine standardisierte Vorgabe, wie die Domains im Fall der Nutzung von MX-Records im Zertifikat aussehen sollen. Es ist daher kein Wunder, dass MTAs oft weder TLS anbieten noch nutzen. Eine Authentifizierung im allgemeinen Fall ist daher bei der MTA-MTA-Kommunikation nicht möglich. Wären diese Probleme gelöst, so blieben noch die Probleme der Sperrung von Zertifikaten und das Problem der kompromittierten CAs. Ersteres ließe sich durch die durchgehende Anwendung von OCSP stapeling mit must-staple lösen, zweiteres hofft Google durch Certificate Transparency in den Griff zu bekommen. Eine Abmilderung der Probleme wäre mit Certificate Pinning möglich; es gibt aber auch hier für den Mailbereich, im Gegensatz zu HTTP ([RFC 7469]), keinerlei Standards.

In jedem Fall muss man feststellen, dass die mit den verschiedenen (Reparatur-)Verfahren verbundenen Algorithmen zur zertifikatsbasierten Authentifizierung komplex und damit fehleranfällig sind. Um diese Probleme zu mindern bzw. ganz zu umgehen, wurde Ende 2010 die Arbeitsgruppe DNS-based Authentication of Named Entities (DANE-WG) der IETF gegründet, die im August 2012 den RFC 6698 publizierte. Auf Basis dieses RFCs werden nun weitere RFCs für den Einsatz im Mailbereich entwickelt.

DANE TLSA und seine Anwendung

DANE, mit Abstützung auf DNSSEC, bietet je nach verwendetem Betriebsmodus eine Erweiterung oder sogar eine Alternative zur althergebrachten PKIX. Hierbei werden Informationen über das zu erwartende Zertifikat direkt beim angesprochenen Hostnamen in einem neu definierten TLSA-Resource-Record im DNS hinterlegt und durch DNSSEC abgesichert, d. h. digital signiert. Der TLSA-Record bindet somit ein Zertifikat an einen Domainnamen. Das Vorhandensein des TLSA-Records signalisiert außerdem, dass grundsätzlich eine mit TLS gesicherte Verbindung aufzubauen ist (Zwangsverschlüsselung). Ein DNSSEC/DANE-fähiger Client kann diese Informationen daher zum Aufbau einer authentifizierten TLS-Verbindung nutzen.

Die DNS-PKI hat eine Reihe von Vorteilen gegenüber der PKIX. Die Signierung ist fest an den DNS-Baum gekoppelt, ein Kind-Knoten kann grundsätzlich nur durch einen Eltern-Knoten signiert werden, womit das fundamentale Problem im CA-Modell – jede CA kann für jede Domain ein Zertifikat ausstellen – wegfällt. DNSSEC ist streng hierarchisch organisiert, sodass es nur wenige Vertrauensanker gibt, die einfach zu verteilen sind. Auch die Erstellung der Signierungskette ist dadurch nicht so kompliziert wie bei der PKIX.

Aufbau eines TLSA-Records

Für jede zu schützende Kombination aus Port, Protokoll (im allgemeinen TCP) und Hostname wird mindestens ein TLSA-Record in die DNS-Zone eingefügt, der die folgenden Informationen enthält:

```
_<PORT>._<PROTOCOL>.<BASE DOMAIN> IN
TLSA <CertificateUsage> <Selector>
<MatchingType> <Data>
```

- *PORT* ist der numerische Port in Dezimalnotation, auf dem der angesprochene Dienst zur Verfügung gestellt wird, beispielsweise 25 für SMTP oder 443 für den Standardport von HTTPS.
- *PROTOCOL* ist das Protokoll der Schicht 4. Im Allgemeinen wird hier TCP Verwendung finden, es ist jedoch auch möglich UDP (Datagramm Transport Layer Security, DTLS RFC 6347) oder SCTP-basierte (Stream Control Transmission Protocoll, RFC 4960) Dienste abzusichern.
- *BASE DOMAIN* wird aus dem Hostname des TLS-Servers bestimmt (s.u.)
- Im Feld *CertificateUsage* ist hinterlegt welche Art Zertifikat zu erwarten ist und wie dieses zu verifizieren ist (s.u.).

- Im Feld *Selector* ist definiert, ob die im Feld *Data* hinterlegten Informationen das vollständige Zertifikat (Full Certificate (Cert), Wert 0) oder nur den öffentlichen Teil des Schlüssels (SubjectPublicKeyInfo (SPKI), Wert 1) beschreiben.
- Im *MatchingType* ist hinterlegt, ob das Feld *Data* nun schlussendlich das volle Zertifikat bzw. die SPKI (Full, Wert 0) oder nur den SHA–256 Hash (Wert 1) bzw. den SHA–512 Hash (Wert 2) davon enthält. Die Nutzung des vollen Zertifikats wird nicht empfohlen, da die daraus resultierenden DNS-Einträge sehr groß werden. Die Auswahl zwischen SHA–256 und SHA–512 wird dem Administrator überlassen, wobei in der freien Wildbahn derzeit SHA–256 klar die Mehrheit darstellt.

Die einzelnen Attribute und deren Verwendungszweck wird in den folgenden Abschnitten erläutert.

Betriebsmodi (CertificateUsage)

In RFC 6698 werden vier Betriebsmodi definiert als Kombination aus Verwendung und Zertifikatsart. Um die sprachliche Verständigung zu erleichtern, wurden durch RFC 7218 Namen für die einzelnen Werte festgelegt. In der folgenden Tabelle werden die Betriebsmode zusammengefasst. In Klammern ist der Attributwert für *CertificateUsage* angegeben.

Tabelle 1: DANE Betriebsmodi.

	zusätzliche PKIX-Validierung	**DANE allein ausreichend**
Trust Anchor	PKIX-TA (0)	DANE-TA (2)
End-Entity Certificate	PKIX-EE (1)	DANE-EE (3)

- PKIX-TA – „CA Constraint“: Mit PKIX-TA wird festgelegt, dass in der PKIX Zertifikatskette ein bestimmtes CA-Zertifikat vorkommen muss. Zur Authentifizierung gelten im Prinzip die üblichen Regeln der PKIX. Bei cross-zertifizierten TA-Zertifikaten muss die jeweilige Kette aber bis zum Ende – meist einem selbstsignierten Wurzelzertifikat – verlängert werden, damit das Zertifikat aus dem TLSA-Record auch sicher gefunden werden kann (s. DANE-OPS). Da mehrere TLSA-Records existieren können, kann auf diese Weise eine eingeschränkte Menge an CAs festgelegt werden, die Zertifikate für eine Target Server Host Domain ausgeben dürfen.
- PKIX-EE – „Service Certificate Constraint“: PKIX-EE gibt an welches EE-Zertifikat der TLS-Server schicken darf. Zusätzlich muss auch hier das Zertifikat die normalen PKIX-Validierungsschritte erfüllen. Es wird somit die Menge an erlaubten EE-Zertifikate eingeschränkt.
- DANE-TA – „Trust Anchor Assertion“: Mit DANE-TA wird ein neuer Vertrauensanker eingeführt. Dies erlaubt den Betrieb einer lokalen CA unabhängig von den öffentlichen CAs. Damit hat man die volle Kontrolle über die Ausgabe neuer Zertifikate. Die Validierung der Zertifikatskette und die Überprüfung des Referenz-Identifiers geschieht wieder mit den Mitteln der PKIX, nur wird hier der TA-Store mit den CA-Zertifikaten nicht mehr genutzt. Damit dies möglich ist, muss der Server immer die volle Kette der Zertifikate einschließlich des TA-Zertifikats mitschicken.
- DANE-EE – „Domain Issued Certificate“: Bei DANE-EE dient das Zertifikat nur noch als Verpackung für den öffentlichen Schlüssel. Es wird weder die Zertifikatskette, noch der Gültigkeitszeitraum oder das Subject des Zertifikats überprüft. Server müssen bei diesem Modus nicht mehr SNI implementieren, auch wenn sie unter verschiedenen Domainnamen bekannt sind, da es möglich ist, die TLSA-Records für die Domainnamen über CNAMEs auf das Default-Zertifikat zu binden.

Vertraut eine Applikation in erster Linie auf die PKIX-Authentifizierung und möchte DANE nur zur zusätzlichen DNSSEC-basierten Einschränkung der möglichen Zertifikate nutzen, so sollten nur die PKIX-Modi und nicht die DANE-Modi genutzt werden. Anderenfalls könnte ein Angreifer, der den TLSA-Record manipulieren kann, von PKIX auf DANE umschalten, in die TLS-Verbindung sein Zertifikat einschleusen und damit die PKIX-Zertifikats-Validierung aushebeln. Umgekehrt, wenn Vertrauen in die DNS-PKI besteht, macht es wenig Sinn zusätzlich zur DANE- auch noch eine PKIX- Überprüfung durchzuführen.

Bestimmung der Base-Domain für den TLSA-Record

Die Base-Domain für die Abfrage des TLSA-Records wird analog zur Bestimmung der Reference-Identifier durchgeführt. Gibt es eine expandierte TSHD, so wird zuerst diese als Base-Domain zur Abfrage verwendet. Findet man unter dieser Domain keinen TLSA-Record so wird die initiale TSHD versucht. Für die Bestimmung der Reference-Identifier hat dies zugleich die Konsequenz, dass nur die

TSHD – initiale oder expandierte – genutzt wird zu der ein TLSA-Record existiert. Die andere fällt weg. Für SNI wird dann die so bestimmte TSHD genommen (so ist es zumindest für SMTP festgelegt, s. Absatz 2.2 von DANE-SMTP).

Austausch bzw. Sperrung von Schlüsseln

Dadurch, dass mehrere TLSA-Records existieren können, ist es möglich ohne Probleme ein Zertifikat mit seinem öffentlichen Schlüssel auszutauschen. Es wird jeweils ein Record für das alte und das neue Zertifikat konfiguriert. Nach Abwarten der zweifachen TTL-Zeit des TLSA-Records kann das Zertifikat ausgetauscht und der Record mit dem alten Zertifikat gelöscht werden. Eine Sperrung eines Zertifikats wird einfach durch den Austausch des Zertifikats erreicht. Es müssen keinerlei zusätzliche Maßnahmen getroffen werden.

Durch den Einsatz von SPKI als Selektor kann bei den Varianten PKIX-EE und DANE-EE das Zertifikat des Servers sogar ohne Anpassung des TLSA-Records erneuert werden, sofern das Schlüsselpaar wiederverwendet wird.

DANE-Betriebsmodi für SMTP

In DANE-SMTP ist festgelegt, dass für SMTP nur die DANE-Modi genutzt werden sollen, da es bisher bei SMTP keine funktionierende PKIX-Authentifizierung gibt, insbesondere fehlt es im Vergleich zum Web an einer anerkannten Menge von Vertrauensankern. Da beim Aufbau der TLS-Verbindung kein Mensch involviert ist, kann auch niemand auf einen OK-Button klicken, falls für die Zertifikatskette kein Vertrauensanker konfiguriert ist.

Empfohlen wird primär die Kombination DANE-EE mit Selektor SPKI und MatchingType SHA–256 einzusetzen, sofern die Koordination bei der MTA- und der DNS-Konfiguration im Falle eines Schlüsselaustausches auf einfache Weise möglich ist. Dies ist der Fall, wenn sowohl der Eigentümer der Service-Domain als auch der Betreiber des Servers DNSSEC einsetzen. Sowohl die TSHD als auch der TLSA-Record liegen dann in einer DNS-Zone beim Server-Betreiber. Wird als Selektor SPKI eingesetzt, so ist eine Erneuerung des Zertifikats ohne Änderung des TLSA-Records möglich. Muss das Schlüsselpaar ausgetauscht werden, hat der Server-Betreiber Zugriff auf den TLSA-Record und das Zertifikat. Da der Server nur sein Default-Zertifikat benötigt, ist auch kein Support für SNI notwendig.

Setzt der Server-Betreiber hingegen kein DNSSEC ein, so müssen der TLSA-Record und die initiale TSHD in der DNS-Zone des Eigentümers der Service-Domain angelegt werden. Zeigt der TLSA-Record auf das Default-Zertifikat des Servers, so muss bei einem Schlüsselaustausch der Serverbetreiber die Änderung des Zertifikats mit allen Eigentümern der Service-Domains koordinieren, was beliebig aufwendig werden kann. Es wird aber ebenfalls kein SNI-Support benötigt. Zeigt der TLSA-Record auf ein Zertifikat des Eigentümers, so muss dieser dem Server-Betreiber das Schlüsselpaar inklusive Zertifikat zukommen lassen und den Austausch koordinieren. In diesem Fall muss SNI unterstützt werden, damit das richtige Zertifikat ausgewählt wird.

Werden viele MTAs betrieben und möchte der Betreiber es sich ersparen, jedes Zertifikat in einem TLSA-Record zu hinterlegen, so kann als Alternative auch DANE-TA verwendet werden. Bei jeder Serverdomain kann dann ein TLSA-Record als CNAME auf einen zentralen TLSA-Record definiert werden. Nur an dieser Stelle muss bei einer Änderung des TA-Zertifikats eine Anpassung geschehen. Auch in diesem Fall muss kein Support für SNI implementiert sein. Die Clients müssen aber im Gegensatz zu DANE-EE die Zertifikatskette einer PKIX-Validierung unterziehen und den Server-Identifier überprüfen.

Bei Nutzung von DANE-TA macht es auch Sinn den Vertrauensanker, der von jedem Server in der Zertifikatskette mitgeschickt werden muss, eindeutig zu identifizieren, damit ein neues TA-Zertifikat nicht auf einmal andere Eigenschaften – insbesondere Einschränkungen – wie vorher hat. Die Empfehlung lautet daher DANE-TA mit Selektor Cert und MatchingType SHA–256 zu verwenden.

TLS ohne Zertifikate (raw keys)

Im Modus DANE-EE schickt der Server eine Zertifikatskette, bei der das EE-Zertifikat als reiner Transportbehälter für die Subject Public Key Info (SPKI) dient. Es macht daher Sinn diesen Overhead zu eliminieren. RFC 7250 definiert eine TLS-Erweiterung bei der Client und Server aushandeln können welche Zertifikatstypen sie unterstützen. Unterstützen beide den Typ RawPublicKey so braucht der Server statt der Zertifikatskette nur noch die SPKI schicken. Ist auch noch die TLS-Erweiterung cached_info (TLS-Cache) implementiert, so reduziert sich das weiter auf den SHA–256 Hash der SPKI.

Wird also wie empfohlen DANE-EE mit Selektor SPKI und MatchingType SHA–256 verwendet, kann der Client diese beiden Erweiterungen nutzen und die Überprüfung des Zertifikats reduziert sich auf den Vergleich der Hashes der SPKI. Damit hat sich TLS vollständig von der PKIX gelöst.

Tabelle 2: Mail-Statisik über fünf Tage im Juli 2015.

TLS-Status/Instanz	mailout	postout	forwout	Gesamt
Kein TLS (Klartext)	37.389 (13,4%)	18.962 (8,4%)	16.256 (2,4%)	72.607 (6,1%)
Anonymous TLS	48.577 (17,4%)	32.692 (14,4%)	35.637 (5,2%)	116.906 (9,8%)
Untrusted (nicht validierbar)	15.297 (5,5%)	10.911 (4,8%)	25.074 (3,7%)	51.282 (4,3%)
Trusted (validierbar)	169.191 (60,7%)	157.081 (69,3%)	556.007 (81,5%)	882.279 (74,3%)
Secure (manuell konfiguriert)	2.495 (0,9%)	2.287 (1,0%)	10.525 (1,5%)	15.307 (1,3%)
DANE	5.816 (2,1%)	4.643 (2,0%)	39.034 (5,7%)	49.493 (4,2%)
Summe	278.765 (100,0%)	226.576 (100,0%)	682.533 (100,0%)	1.187.874 (100,0%)

Koordination zwischen Mail- und DNS-Administration

In den wenigsten Fällen werden Mail- und DNS-Server von den selben Personen betrieben. Dadurch kann es passieren, dass der TLSA-Record und das Server-Zertifikat nicht übereinstimmen, bedingt durch Fehler der jeweiligen Administratoren oder Problemen in der Koordination. Daher ist es unbedingt notwendig, dass beim Monitoring der Systeme auch der TLSA-Record mit überwacht wird, um Fehler oder Manipulationen zu entdecken. Zu diesem Zweck können Monitoring-Plugins wie check_posttls_finger verwendet werden. Eine andere einfache Möglichkeit ist die konsequente interne Nutzung von DANE, die sich im Fehlerfall durch lokale Fehlermeldungen und entsprechend konfigurierte Trigger im Monitoringsystem bemerkbar macht.

DANE TLSA für Mailserver in der Praxis

Am Mailsystem des Leibniz-Rechenzentrum ist DANE ausgehend auf einigen Teilsystemen bereits seit Februar 2014 produktiv in Verwendung. Die verwendete Software Postfix unterstützt diesen Modus seit Version 2.11, die im Januar 2014 veröffentlicht wurde. Sie benötigt als Voraussetzung eine halbwegs aktuelle OpenSSL-Library und eine DNSSEC-fähige Resolver-Library. Beide Bedingungen waren in der verwendeten Distribution Debian Wheezy erfüllt, so dass der Installation der Software nichts im Wege stand. Seit April 2014 steht im *wheezy-backports*-Repository eine aktuelle Version zur Verfügung.

Da ein DNSSEC-validierender Resolver schon zur Verfügung stand waren nur die folgenden Konfigurationsoptionen nötig:

```
smtp_dns_support_level = dnssec
# smtp_tls_security_level = may
smtp_tls_security_level = dane
```

Der Security-Level *dane* schaltet opportunistisches DANE an. Hierbei wird, konform zum IETF-Draft, der DNSSEC-Validierungsstatus der DNS-Antworten geprüft und bei einem vorhandenen, sicheren TLSA-Record von opportunistic TLS auf mandatory TLS mit der signalisierten Zertifikatsüberprüfung umgeschaltet. Ein MITM-Angriff oder gar eine (versehentliche oder erzwungene) Umschaltung auf unverschlüsselte Übertragung ist damit ausgeschlossen.

Seit Ende 2014 wird nun der gesamte ausgehende Mailverkehr des LRZ über DANE-fähige Server abgewickelt. Eine beispielhafte Statistik über fünf Tage Mailvolumen (im Juli 2015, s. Tabelle 2) in drei Instanzen zeigt, dass der Anteil der DANE-gesicherten Mails sich mittlerweile bei rund 4% eingependelt hat. Die Instanzen *mailout* und *postout* sind größtenteils für den Verkehr von MWN-Benutzern an externe Empfänger zuständig, wobei sich mailout um die E-Mails der lokalen Mailserver im MWN und postout um die der zentralen Mailsysteme des LRZ kümmert. Die Instanz *forwout*, die für die Weiterleitungen der E-Mails der zentralen Systeme sorgt, zeigt sogar einen deutlich höheren Anteil. Dies liegt daran, dass ein nicht unerheblicher Anteil der weitergeleiteten E-Mails an Mailserver im MWN geht, die – bedingt durch die starke Propagierung von DNSSEC innerhalb des MWNs – bereits mit DANE abgesichert wurden.

Um diese jedoch adäquat anbinden zu können war es nötig, auf den Mailservern validierende Resolver (Unbound) zu installieren, da aus historischen Gründen die

Zonen der meisten MWN-Teilnehmer als Slave auf den offiziellen Resolvern (BIND) vorgehalten werden und daher nicht dort validiert werden können. Durch die Nutzung des Systemkonfigurationswerkzeugs *Puppet* ist die Konfiguration der lokalen Resolver mit keinem Mehraufwand verbunden.

- Bei Verbindungen ohne TLS kommt keine Verschlüsselung zustande. Üblicherweise geschieht dies aufgrund von fehlendem STARTTLS-Support auf der Gegenstelle. Es könnte sich jedoch auch um eine Downgrade-Attacke handeln, eine Bewertung ist höchstens durch den Vergleich mit älteren Verbindungen zum gleichen Host möglich.
- Anonymous TLS ist eine Gruppe von Cipher-Suites, bei der keine Zertifikate ausgetauscht werden (da sie nicht validiert werden entstehen dadurch auch keine Nachteile). Sie kommen insbesondere zwischen zwei Postfix-Instanzen zum Einsatz, wenn beide Seiten signalisieren keine Zertifikatsprüfung durchführen zu wollen.
- Bei den Untrusted-Sessions wurde vom Gegenüber ein Zertifikat übermittelt, bei dem aber die klassische PKIX-Validierung fehlschlägt. Dabei handelt es sich in den meisten Fällen um selbstsignierte oder abgelaufene Zertifikate.
- Bei Trusted-Sessions wurde hingegen ein validierbares Zertifikat präsentiert (am LRZ kommt hierbei der systemseitige TA-Store von Debian zum Einsatz). Obwohl der Anteil vergleichsweise hoch ist (hauptsächlich bedingt durch die weite Verbreitung von offiziellen Zertifikaten bei den großen ISPs) sieht man die Verlustrate, die man bei der Umstellung auf zwangsweise Verschlüsselung mit korrekten Zertifikaten analog zum Web in Kauf nehmen müsste.
- Bei Secure-Sessions handelt es sich um manuell konfigurierte Trust-Anchors für lokale Systeme, wie im vorherigen Abschnitt “Certificate Pinning im Münchner Wissenschaftsnetz (MWN)” beschrieben.
- DANE schlussendlich sind Verbindungen, deren Zertifikat gemäß dem beschriebenen Protokoll mit DNSSEC und TLSA-Records abgesichert sind.

Zur Absicherung des eingehenden Mailverkehrs mit DANE ist keinerlei spezielle Unterstützung in der MTA-Software nötig, sofern STARTTLS unterstützt wird. Allerdings müssen, wie zuvor beschrieben, sowohl die Service-Domain (Maildomain) als auch die Target Server Host Domain (TSHD) innerhalb einer DNSSEC-signierten und sicher delegierten Zone liegen und TLSA-Records für die MTAs im DNS hinterlegt werden.

Dies zog sich am LRZ etwas länger hin, da zuerst die Hauptzone des LRZ (lrz.de) mit DNSSEC signiert werden musste. Aufgrund der enormen Wichtigkeit dieser Zone für alle durch das LRZ erbrachten Dienste musste diese Änderung gut abgestimmt werden. Zusätzlich musste noch die Fähigkeit zur Veröffentlichung von TLSA-Records durch den Hersteller der DNS-Verwaltungsplattform nachgerüstet werden, was zum Glück innerhalb weniger Wochen möglich war. Seit Oktober 2014 sind für die beiden MTA-Cluster des LRZ (mailrelay und postrelay, mit jeweils zwei Hosts) TLSA-Records veröffentlicht. Dazu wird der Quasi-Standard eines DANE-EE Eintrags mit dem öffentlichen Schlüssel (SPKI, Selektor 1) und einem SHA256-Hash verwendet. Ein TLSA-Eintrag im DNS kann nun beispielhaft folgendermaßen aussehen:

```
_25._tcp.postrelay1.lrz.de. 86400 IN TLSA 3 1
1 3D805975A26979F67B8F35AD20DC245B96FF6B03
16ACE7CE08BFEFA0 C9BC021E
```

Dieser definiert, dass auf Port 25/TCP (im Allgemeinen als SMTP bezeichnet) auf dem Host postrelay1.lrz.de ein EE-Zertifikat erwartet werden soll (DANE-EE, *Certificate Usage* = 3), in dem der öffentliche Teil des Schlüssels (Selektor = 1 , SPKI) einen SHA–256 Hash (*MatchingType* = 1, SHA–256) von 3D80... besitzt.

Ein DANE-fähiger SMTP-Client braucht somit keine PKIX-Validierung des Zertifikats mehr durchführen, eine DANE-basierte Validierung allein ist ausreichend. Da das LRZ als Zertifikate aber ganz normale Zertifikate aus der DFN-PKI und keine selbstsignierten einsetzen, kann ein nicht-DANE-fähiger Client die normale PKIX-Validierung nutzen. Somit können für eine Übergangszeit beide Welten zufrieden gestellt werden.

Da innerhalb des TLS-Protokolls keine Signalisierung eines DANE-Status erfolgt, sind genaue Statistiken über die Nutzung von DANE bei ankommenden E-Mails naturgemäß nicht möglich. Eine (durch Cachingeffekte im DNS eher zu kleine) Abschätzung ist die Anzahl der DNS-Anfragen nach den TLSA-Records. Hier sehen wir derzeit knapp 1000 Anfragen pro Tag, was grob auf eine niedrige vierstellige Anzahl von mit DANE gesicherten TLS-Verbindungen schließen lässt.

Probleme mit DANE in der Praxis

Selbst in den Anfangszeiten von DANE im LRZ traten nur sporadische Probleme auf, die alle auf Konfigurationsfehler der Gegenstelle zurückgeführt werden konnten. Ein beliebter Fehler bei authoritativen Servern, der aber leider hauptsächlich im Zusammenhang mit DANE bemerkt wird, ist eine falsche Logik bei der Übermittlung der NSEC(3)-Records, wenn kein TLSA-Eintrag existiert. Insbesondere in Zusammenhang mit Wildcard-Records auf Zonen-

ebene sind beispielsweise einige djbdns-Versionen anfällig für falsche Antworten, welche auf dem Resolver einen Validierungsfehler hervorrufen. Dies äußert sich darin, dass der MX- und A-Record zwar aufgelöst werden kann, der TLSA-Record in der gleichen Zone jedoch nicht. Da ein validierender Resolver prinzipbedingt keine Unterscheidung zwischen einem Serverfehler und einem Angriff treffen kann (und dies den sendenden MTA auch auf gar keinen Fall zu einem Rückfall zur unverschlüsselten Übertragung bringen darf) bleibt die Mail in diesen Fällen mit einer Fehlermeldung in der Queue liegen.

Mit der Postfix-Konfigurationsoption *smtp_tls_policy_maps* kann der global auf *dane* gesetzte Security-Level pro Zieldomain überschrieben und so beispielsweise auf *may* (opportunistisches TLS ohne DANE) gesetzt werden, wenn es zu Fehlern bei der Auflösung des TLSA-Records kommt. Zusammen mit einem Trigger in Monitoringsystem auf die Symptome eines TLSA-Lookupfehlers kann die Mailzustellung dennoch sichergestellt werden. Im Laufe des Jahres betraf dies jedoch nur etwa ein Dutzend Domains, da V. Dukhovni, der DANE für Postfix implementiert hat, sehr viele fehlerhafte Konfiguration melden und beheben lassen konnte.

Weitere typische Probleme, die in der Praxis auftreten können, werden auf der Seite https://dane.sys4.de/common_mistakes beschrieben.

Handlungsempfehlungen

Wie bereits im Beitrag zu DNSSEC empfohlen ist der Einsatz von DNSSEC-validierenden Resolvern generell angeraten. Da durch die Validierung alle mit DNSSEC geschützten DNS-Einträge gegen Spoofing geschützt sind (selbst wenn die Clientsoftware keinerlei DNSSEC-Fähigkeiten mitbringt) bringt dies einen sofortigen Sicherheitszuwachs mit sich.

Mit einer validierenden Resolverinfrastruktur können die Mailserver mit DANE-Support (Postfix 2.11+ oder Exim 4.85+ (experimentell)) auf opportunistic DANE umgeschaltet werden. Dadurch wird verifiziertes TLS bei ausgehenden Verbindungen zu DANE-fähigen Zielen erzwungen und eine unverschlüsselte Übertragung verhindert. Dies sollte jedoch, wie in jedem IT-Projekt üblich, durch Tests und Monitoring flankiert werden.

Es wird oft empfohlen, einen lokalen validierenden Resolver (beispielsweise Unbound) auf jedem Host einzusetzen. Rein technisch ist dies nicht nötig, wenn ein absolut sicherer, gegen Spoofing geschützter Weg zwischen dem Mailserver und dem Resolver existiert. Andernfalls könnte ein Angreifer das *ad*-Flag (authenticated data), das die erfolgreiche Authentifizierung signalisiert, in der DNS-Antwort nach Belieben verändern. Diese Garantie ist, je nach (Netz-)struktur, nur schwer zu geben, so dass die generische Empfehlung zu einem lokalen, validierenden Resolver in den meisten Fällen Gültigkeit hat. Zur Verbesserung der Caching-Rate (oder zur Nutzung lokaler Zonen) kann dieser seine Anfragen zum netzweiten Resolver weiterleiten (sofern dieser DNSSEC-fähig ist).

Auf der empfangenden Seite benötigt DANE – abgesehen von STARTTLS – keine Unterstützung in der MTA-Software, diese Technologie kann daher auch bei anderen Plattformen wie Exchange oder kommerziellen Appliances eingesetzt werden. Die Veröffentlichung von TLSA-Records ist daher für jeden Betreiber zu empfehlen, da damit eine Zwangsverschlüsselung signalisiert wird. Allerdings ist DANE nur im Zusammenhang mit DNSSEC-signierten Zonen sinnvoll. Die Signierung einer Zone und die damit verbundenen Fragestellungen wurden bereits in DNSSEC detailliert beleuchtet.

Fazit

Die Transportverschlüsselung von E-Mails hat aufgrund der beim E-Mail-Versand anfallenden Metadaten eine langfristige Berechtigung und ist in Zeiten noch seltener Ende-zu-Ende-Verschlüsselung von E-Mail-Inhalten in der Regel der einzige Ansatz, die Vertraulichkeit von E-Mails zumindest partiell sicherzustellen. Legt man das in der Post-Snowden-Ära realistische Szenario zugrunde, dass ein Angreifer aktiv in Kommunikationsvorgänge eingreifen kann, so zeigt sich, dass herkömmliche Ansätze wie STARTTLS für SMTP unzuverlässig sind. Zugleich ist E-Mail-Versand für die Betreiber von Mailservern immer Massenbetrieb in dem Sinn, dass eine manuelle Kontrolle einzelner Verbindungen und dabei präsentierter Peer-Zertifikate nicht ohne praktisch zu hohen Aufwand möglich und in MTA-Software auch technisch gar nicht vorgesehen ist.

DANE TLSA ist der erste standardisierte Ansatz, die zum Aufbau authentifizierter und verschlüsselter Verbindungen erforderlichen Metadaten über DNS zugänglich zu machen. Beim zugrundegelegten Angreifermodell ist eine Absicherung dieser Informationen über DNSSEC zwingend erforderlich. Während DANE TLSA beispielsweise für den Einsatz im Zusammenspiel mit HTTPS bei den großen Herstellern von Webbrowsern bislang nur auf wenig Interesse gestoßen ist, unterstützen es mehrere, weit verbreitete MTA-Produkte bereits für den praktischen Einsatz ausreichend stabil. Auf Basis einer bereits vorhandenen DNSSEC-Infrastruktur stellt die Anpassung der eigenen Mailserver an DANE TLSA einen sehr überschaubaren Auf-

wand dar. Wie das Beispiel der LRZ-Mailserver zeigt, lässt sich mit relativ geringem Aufwand bereits ein signifikanter Teil des Mailverkehrs absichern, obwohl DNSSEC erst noch vor seinem großen Durchbruch steht. Wenn beispielsweise die Universitäten, Hochschulen und der DFN-Verein DANE und DNSSEC für ihre Haupt-(Mail-)Domains konfigurieren würden, könnte sehr einfach der nationale Mailaustausch im Wissenschaftsbereich abgesichert werden. Insgesamt ist zu erwarten, dass sich DANE TLSA, mindestens im Umfeld von Mailservern, in den nächsten Jahren als breit akzeptierter Standard durchsetzen wird.

Literatur und Referenzen

BayLDA-SMTP — Onlineprüfung bei Mailservern hinsichtlich STARTTLS, Perfect Forward Secrecy und Heartbleed-Lücke, September 2014. http://www.lda.bayern.de/onlinepruefung/emailserver.html

BSI TR 01201 Teil 1.4 — TR – De-Mail IT-Basisinfrastruktur Interoperabilitätsspezifikation, Februar 2014. https://www.bsi.bund.de/SharedDocs/Downloads/DE/BSI/Publikationen/TechnischeRichtlinien/De_Mail/TR_De_Mail_IT-Binfra_IO_pdf.pdf?__blob=publicationFile

CAB-Forum — CA/BROWSER FORUM. https://cabforum.org/

check_posttls_finger — nagios-plugins/check_posttls_finger. https://github.com/benningm/nagios-plugins/blob/master/check_posttls_finger

CloudFlare-Cost — Prince, M.: The Hidden Costs of Heartbleed, April 2014. https://blog.cloudflare.com/the-hard-costs-of-heartbleed/

CRLSet — Langley, A.: No, don't enable revocation checking, April 2014. https://www.imperialviolet.org/2014/04/19/revchecking.html

Draft CT — Laurie, B., Langley, A., Kasper, E., Messeri, E., Stradling, R.: IETF-Draft – Certificate Transparency, draft-ietf-trans-rfc6962-bis–08, July 2015. https://tools.ietf.org/html/draft-ietf-trans-rfc6962-bis-08

DANE-OPS — Dukhovni, V., Hardaker, W. : IETF-Draft – Updates to and Operational Guidance for the DANE Protocol, draft-ietf-dane-ops–13, July 2015. https://tools.ietf.org/html/draft-ietf-dane-ops-13

DANE-SMTP — Dukhovni, V., Hardaker, W. : IETF-Draft – SMTP security via opportunistic DANE TLS, draft-ietf-dane-smtp-with-dane–19, May 2015. https://tools.ietf.org/html/draft-ietf-dane-smtp-with-dane-19

DANE-SRV — Finch, T, Miller, M., Saint-Andre, P.: IETF-Draft – Using DNS-Based Authentication of Named Entities (DANE) TLSA Records with SRV Records, draft-ietf-dane-srv–14, April 2015. https://tools.ietf.org/html/draft-ietf-dane-srv-14

DANE-WG — IETF Working Group: DNS-based Authentication of Named Entities (dane). https://datatracker.ietf.org/wg/dane/documents/

DNSSEC — Feuchtinger, D., Hommel, W., Reiser, H., Schmidt, B., Storz, M.: DNSSEC – Konzepte und Betriebsaspekte des Domain Name Systems der Zukunft, PIK 2015, Heft 1–2, S. 71–81.

djbdns — Bernstein, D. J., djbdns — Domain Name System Tools. http://cr.yp.to/djbdns.html

google-transparenz — Google Transparenzbericht – E-Mail-Verschlüsselung bei der Übertragung, Juni 2015. http://www.google.com/transparencyreport/saferemail/?hl=de

heartbleed — The Heartbleed Bug, April 2014. http://heartbleed.com/

mailbox.org — SSL-Verbindungen vorab geprüft – mailbox.org informiert schon vor E-Mail-Versand über Sicherheitslevel der Empfänger, 2015. https://mailbox.org/mailbox-org-informiert-vor-e-mail-versand-ueber-sicherheitslevel-der-empfaenger/

Marlinspike 2009 — Marlinspike, M.: More Tricks For Defeating SSL In Practice, Blackhat USA 2009, Las Vegas, July 2009. http://www.blackhat.com/presentations/bh-usa-09/MARLINSPIKE/BHUSA09-Marlinspike-DefeatSSL-SLIDES.pdf

Must-Staple — Hallam-Baker, P.: IETF-Draft – X.509v3 TLS Feature Extension, draft-hallambaker-tlsfeature–10, July 2015. https://tools.ietf.org/html/draft-hallambaker-tlsfeature-10

Netcraft-Heartbleed — Mutton, P.: Certificate revocation: Why browsers remain affected by Heartbleed, April 2014. http://news.netcraft.com/archives/2014/04/24/certificate-revocation-why-browsers-remain-affected-by-heartbleed.html

RFC 2595 — Newman, C.: RFC 2595 – Using TLS with IMAP, POP3 and ACAP, June 1999. https://tools.ietf.org/html/rfc2595

RFC 3207 — Hoffman, P.: RFC 3207 – SMTP Service Extension for Secure SMTP over Transport Layer Security, February 2002. https://tools.ietf.org/html/rfc3207

RFC 4409 — Gellens, R., Klensin, J.: RFC 4409 – Message Submission for Mail, April 2006. https://tools.ietf.org/html/rfc4409

RFC 4616 — Zeilenga, K.: RFC 4616 – The PLAIN Simple Authentication and Security Layer (SASL) Mechanism, August 2006. https://tools.ietf.org/html/rfc4616

RFC 4960 — Stewart, R. (Ed.), RFC 4960 — Stream Control Transmission Protocol, September 2007. https://tools.ietf.org/html/rfc4960

RFC 4985 — Santesson, S: RFC 4985 – Internet X.509 Public Key Infrastructure Subject Alternative Name for Expression of Service Name, August 2007. https://tools.ietf.org/html/rfc4985

RFC 5246 — Dierks, T, Rescorla, E.: RFC 5246 – The Transport Layer Security (TLS) Protocol Version 1.2, August 2008. https://tools.ietf.org/html/rfc5246

RFC 5280 Cooper, D., Santesson, S., Farrell, S., Boeyen, S., Housley, R., Polk, W.: RFC 5280 – Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile, May 2008. https://tools.ietf.org/html/rfc5280

RFC 6066 Eastlake, D.: RFC 6066 – Transport Layer Security (TLS) Extensions: Extension Definitions, January 2011. https://tools.ietf.org/html/rfc6066

RFC 6125 Saint-Andre, P., Hodges, J.: RFC 6125 – Representation and Verification of Domain-Based Application Service Identity within Internet Public Key Infrastructure Using X.509 (PKIX) Certificates in the Context of Transport Layer Security (TLS), March 2011. https://tools.ietf.org/html/rfc6125

RFC 6186 Daboo, C.: RFC 6186 – Use of SRV Records for Locating Email Submission/Access Services, March 2011. https://tools.ietf.org/html/rfc6186

RFC 6347 Rescorla, E., Modadugu, N.: RFC 6347 — Datagram Transport Layer Security Version 1.2, Januar 2012. https://tools.ietf.org/html/rfc6347

RFC 6376 Crocker, D., Hansen, T., Kucherawy, M.: RFC 6376 – DomainKeys Identified Mail (DKIM) Signatures, September 2011. https://tools.ietf.org/html/rfc6376

RFC 6698 Hoffman, P., Schlyter, J.: RFC 6698 – The DNS-Based Authentication of Named Entities (DANE) Transport Layer Security (TLS) Protocol: TLSA, August 2012. https://tools.ietf.org/html/rfc6698

RFC 6797 Hodges, J., Jackson, C., Barth, A.: RFC 6797 – HTTP Strict Transport Security (HSTS), November 2012. https://tools.ietf.org/html/rfc6797

RFC 6960 Santesson, S., Myers, M., Ankney, R., Malpani, A., Galperin, S., Adams,C.: RFC 6960 – X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP, June 2013. https://tools.ietf.org/html/rfc6960

RFC 6961 Pettersen, Y.: RFC 6961 – The Transport Layer Security (TLS) Multiple Certificate Status Request Extension, June 2013. https://tools.ietf.org/html/rfc6961

RFC 6962 Laurie, B., Langley, A., Kasper, E.: RFC 6962 – Certificate Transparency, June 2013. https://tools.ietf.org/html/rfc6962

RFC 7208 Kitterman, S.: RFC 7208 – Sender Policy Framework (SPF) for Authorizing Use of Domains in Email, Version 1, April 2014. https://tools.ietf.org/html/rfc7208

RFC 7218 Gudmundsson, O.: RFC 7218 – Adding Acronyms to Simplify Conversations about DNS-Based Authentication of Named Entities (DANE), April 2014. https://tools.ietf.org/html/rfc7218

RFC 7250 Wouters, P., Tschofenig, H., Gilmore, J., Weiler, S., Kivinen, T.: RFC 7250 – Using Raw Public Keys in Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS), June 2014. https://tools.ietf.org/html/rfc7250

RFC 7258 Farrell, S., Tschofenig, H.: RFC 7258 – Pervasive Monitoring Is an Attack, May 2014. https://tools.ietf.org/html/rfc7258

RFC 7435 Dukhovni, V.: RFC 7435 – Opportunistic Security: Some Protection Most of the Time, December 2014. https://tools.ietf.org/html/rfc7435

RFC 7457 Sheffer, Y., Holz, R., Saint-Andre, P.: RFC 7457 – Summarizing Known Attacks on Transport Layer Security (TLS) and Datagram TLS (DTLS), Februar 2015. https://tools.ietf.org/html/rfc7457

RFC 7469 Evans, C., Palmer, C., Sleevi, R.: RFC 7469 – Public Key Pinning Extension for HTTP, April 2015. https://tools.ietf.org/html/rfc7469

RFC 7489 Kucherawy, M., Zwicky, E.: RFC 7489 – Domain-based Message Authentication, Reporting, and Conformance (DMARC), March 2015. https://tools.ietf.org/html/rfc7489

RFC 7525 Sheffer, Y., Holz, R., Saint-Andre, P.: RFC 7525 – Recommendations for Secure Use of Transport Layer Security (TLS) and Datagram Transport Layer Security (DTLS), May 2015. https://tools.ietf.org/html/rfc7525

TACK Perrin, T., Marlinspike, M.: Draft – Trust Assertions for Certificate Keys, draft-perrin-tls-tack–02, January 2013. https://tools.ietf.org/html/draft-perrin-tls-tack-02

TLS-Cache Santesson, S., Tschofenig, H.: IETF-Draft – Transport Layer Security (TLS) Cached Information Extension, draft-ietf-tls-cached-info–19, March 2015. https://tools.ietf.org/html/draft-ietf-tls-cached-info-19

Daniel Feuchtinger: Leibniz Supercomputing Centre, Garching n. Munich, Bavaria, Germany

Wolfgang Hommel: Leibniz Supercomputing Centre, Garching n. Munich, Bavaria, Germany

Helmut Reiser: Leibniz Supercomputing Centre, Garching n. Munich, Bavaria, Germany

Michael Storz: Leibniz Supercomputing Centre, Garching n. Munich, Bavaria, Germany

Bernhard Schmidt: Leibniz Supercomputing Centre, Garching n. Munich, Bavaria, Germany