Bachelor Degree Project

UNIVERSITY OF SKÖVDE

ESTABLISHING DANE TLSA DEPLOYMENT LEVELS AMONG SWEDISH SECOND LEVEL DOMAINS

Bachelor Degree Project in Information Technology
with a Specialisation in Network and
System Administration G2E (IT604G)
Level ECTS
Spring term 2017

Rikard Sandelin
a14riksa@student.his.se

Supervisor: Johan Zaxmy
Examiner: Jonas Gamalielsson

Abstract

Domain Based Authentication of Named Entities (DANE) is an Internet Engineering Task Force (IETF) standard released in 2012 intended to complement or in some cases replace the current Public Key Infrastructure (PKI) model. The current PKI model uses Transport Layer Security (TLS) certificates issued by Certificate Authorities (CA) binding domain names to public key. These CAs act as trust anchors during the certificate validation process. Web browsers and other TLS supported applications have large lists of trusted CA public keys. If one of these trusted CAs are compromised the whole system is compromised. DANE uses the Domain Name System (DNS) to publish TLS certificate information and create certificate associations to domain names. DANE relies on DNS Security Extensions (DNSSEC) for authentication and message integrity. Using the DNS root as a single trust anchor instead of the many CA trust anchors the attack surface is drastically reduced.

In this study a quantitative survey among Swedish DNSSEC signed Second Level Domains (SLD) is performed with the aim to establish the DANE TLSA deployment level among the SLDs in Top Level Domain (TLD) .se.

The results show that 686 471 of the Swedish SLDs have been DNSSEC signed which is approximately 49% of all Swedish SLDs. The number of domains that have deployed DANE is very low, with only 79 SLD found to have DANE TLSA resource records in DNS. The total number of DANE TLSA resource records were 175 and the most common service used with DANE TLSA was HTTPS on port 443 which was 62% of all DANE TLSA resource records found. The most common certificate usage field setting was three, domain issued certificates.

Keywords: DNS, DNSSEC, DANE, Deployment.

Sammanfattning

Domän Baserad Autentisering av Namngivna Entiteter (eng. Domain Based Authentication of Named Entities, DANE) är en *Internet Engineering Task Force* (IETF) standard som publicerades 2012 med avsikten att komplettera och i vissa fall ersätta den nuvarande *Public Key Infrastructure* (PKI) modellen. Den nuvarande PKI modellen använder transportlagersäkerhets (eng. Transport Layer Security, TLS) certifikat utfärdade av så kallade certifikatutfärdare (Certificate Authorities, CA) som binder domännamn till en CAs publika nyckel. Dessa CA:s fungerar som förtroendeankare under valideringsprocessen då certifikatens giltighet valideras. Webbläsare och andra TLS applikationer har ofta stora listor över betrodda CA nycklar. Om en av dessa betrodda CA:s komprometteras så är kan alla domäner signerade av den certifikatutfärdaren vara komprometterade. DANE använder domännamnsystemet (DNS) för att publicera TLS certifikats information och skapa associationer mellan certifikat och domännamn. DANE förlitar sig på DNS *Security Extensions* (DNSEC) för autentisering och meddelandeintegritet. Genom att bara ha DNS roten som förtroendeankare istället för de många olika förtroendeankarna bland den stora mängden CA:s minskar attackytan avsevärt.

I den här studien genomförs en kvantitativ undersökning bland svenska DNSSEC signerade domäner med syftet att kartlägga implementations graden av DANE TLSA bland domäner under den svenska toppnivådomänen .se.

Resultatet visar att 686 471 svenska domäner har blivit DNSSEC signerade vilket motsvarar 49% av alla svenska domäner. Antalet domäner som har implementerat DANE är väldigt lågt, endast 79 stycken domäner hittades som hade DANE TLSA resursposter i DNS. Det totala antalet DANE TLSA resursposter som hittades uppgick till 175 stycken och den vanligaste tjänsten som användes tillsammans med DANE var HTTPS över port 443, vilket var 62% av samtliga DANE TLSA resursposter som hittades. Den vanligaste typen av TLS certifikat som specificerades i *Certificate Usage Field* i DANE TLSA resursposterna var domänutfärdande certifikat.

Nyckelord: DNS, DNSSEC, DANE, Deployment.

Popular science summary

As more people start using services that requires personal and private information to be sent over the Internet, the need for secure and encrypted communication have increased. Examples of information that should be sent over a secure connection could be usernames and passwords or information related to private financial transactions. When visiting a secure website using a web browser the Hypertext Transport Protocol Secure (HTTPS). In the web browsers address bar the address should start with HTTPS which would indicate that the connection is secure and encrypted. Before a web browser can establish a secure connection to a website it needs to confirm that the website is in fact the correct website and not a fraudulent website posing as the real one designed to steal personal information. Websites prove their identity by presenting a Transport Layer Security (TLS) certificate to the web browser. TLS is used by HTTPS to establish and negotiate the setup of the encrypted connection between the website and the web browser. These certificates contain the domain name of the domain for which the certificate was issued as well as the digital signature of a trusted third-party vouching for the authenticity of the web site. These trusted third parties are called Certificate Authorities (CA). There are hundreds of individual CAs that can sign certificates and there are no restrictions in place for which domains they may issue certificates for. Web browsers usually contain lists of CAs that are trusted by the web browser and this list is used to validate the certificates presented by websites and if a website presents a certificate that is trusted by the web browser the web browser will proceed to establish an encrypted connection to the web site.

The problem with the system using CA signed certificates is that it is dependent on each CA to do thorough checks of any person requesting a signed certificate and make sure that the person is in fact the owner or authorized to request certificates for the domain. If only one of all the CAs that are trusted by for example a web browser issues a bad certificate, all domains signed by that CA may be compromised. A badly issued certificate could be used by an attacker to make a fraudulent web site appear legitimate. A web browser would for example trust a fake Internet bank web site if the web site presents an otherwise valid certificate assuming the issuing CA is trusted by the web browser.

To improve the security and methods by which certificates could be validated the Internet Engineering Task Force (IETF) released a new protocol standard in 2012 called Domain Based Authentication of Named Entities (DANE). DANE provides a way to publish information about certificates and how to validate them within the Domain Name System (DNS). The Domain Name System allows computers communicating over a network such as the Internet to translate domain names to the corresponding Internet Protocol (IP) address of the website they wish to visit. When computers communicate over a network they do so by using their unique IP address. The IP address is usually represented as four fields separated by dots where each field contains a decimal number between 0—255, an example of an IP address is 172.16.254.1. The Domain Name system was developed to make it easier for people to interact with computers by using names rather than forcing people to memorize the unique IP address of each website or service they wish to use.

The Domain Name System have a few security issues. The first problem is that there is no mechanism to verify that the Domain Name System server responding to a Domain Name System query is in fact the correct server authorized to provide responses for the specified domain. The second problem is that it is not possible to detect if a response message has been intercepted and altered between the sender and the recipient by an attacker. To increase the security the Domain Name System Security Extension (DNSSEC) was developed. DNSSEC introduces digital signatures by the use of public key encryption. Public key encryption is based on the notion that there are two encryption keys, one private key that is kept secret by the owner and a public key that is made public and any message encrypted with one of

these keys can only be decrypted by the use of the other key. This means that anyone can use the public key to verify a digital signature generated by the private key which is kept secret by the owner. All the information stored in a Domain Name System server and that will be used to answer Domain Name System queries will have a digital signature as well as the public key required to verify these signatures. This way it is possible to verify that the responses are in fact from the correct server and that the received messages has not been intercepted and altered.

The DANE protocol relies on DNSSEC for security and any computer connecting to a website that use DANE to publish certificate information in DNS can not only verify that the DNS server responding is the correct one, it can verify that any responses from that server has not been altered which means that the IP address provided for whichever service requested is most likely the correct one. Similarly, the information related to the certificate received through DNS can be validated and then compared to the actual certificate presented by the webserver. If for any reason the certificate does not match the information provided through DNS the browser should abort the connection process as it possible the website is insecure and potentially fraudulent.

Throughout this text the example of websites and web browsers have been used, however DANE is not limited to just those cases. Any service that use TLS certificates could theoretically use DANE. Examples of other such services would be e-mail and Instant Messaging (IM). However, during the study it was made clear that there are still very few applications that have support for DANE out of the box. For example, there are currently no web browsers with built in support for DANE. To gain this functionality, users have to rely on third-party extensions.

Since DANE is a relatively new standard, the number of studies related to DANE are still limited in numbers. There are however some studies examining different aspects of DANE such as performance, security and deployment levels among domains. This study is based on one of the previous studies by Zhu et al. (2015). that measured the deployment levels of DANE among .com and .net domains. This study examined the DANE deployment levels among Swedish .se domains. The results of this study could be useful for companies or organizations that either have plans to implement DANE for their own services such as e-mail and web sites. Other organizations that may benefit from the result of this study could be software developers that would like to know if there is a demand for DANE supported applications and services before committing any resources to developing such software solutions.

In this study, a survey among the Swedish .se domains was performed and the survey focused on three types of Internet based services known to support DANE, Web sites using HTTPS, e-mail and Instant Messaging services.

The result shows that the deployment levels among the Swedish .se domains are very low, with only 79 domains using DANE which is less than 0.1% of all Swedish domains. This result is similar to the result of the previous study by Zhu et al. (2015) which examined the .com and .net domains. The study also shows that 49% or 686 471 of all Swedish domains use DNSSEC to increase the security of DNS. This is a substantial number of domains that could correctly implement DANE to improve security for the users of their services. The study also shows that websites using HTTPS is the most common service for which DANE is used. The most common certificate used with DANE are domain issued certificates which is a certificate issued by the domain holders directly without the involvement by a third-party certificate authority's.

The result shows that the deployment levels of DANE are still very low which confirms the findings of previous studies. This indicates that the number of users that would benefit from for example web browsers with build in DANE support is limited, by extension software developers lack incentives to

devote resources to the development of such applications. For other services where support for DANE is already present, the low deployment levels shows that the number of users that would benefit from an organization implementing DANE for their services is low. In order for DANE to gain some traction, it would be beneficial if some larger organizations such as e-mail and Internet service providers were to conduct some large-scale deployments of DANE.

Populärvetenskaplig sammanfattning

Allt eftersom fler människor använder tjänster som kräver att personlig och privat information skickas över Internet har behovet av säker och krypterad kommunikation ökat. Några exempel på information som bör skickas över en säker anslutning är användarnamn och lösenord samt privata finansiella transaktioner. När en webbsida besöks som använder en säker anslutning används *Hypertext Transfer Protocol Secure* (HTTPS). För att se om en anslutning till en webbsida är säker bör adressen som står i webbläsarens adressfält börja med HTTPS, detta indikerar att anslutningen är säker och krypterad. Innan webbläsaren kan etablera en säker anslutning behöver webbläsaren bekräfta att webbsidan är den korrekta och inte en bedräglig sida som imiterar den riktiga för att stjäla personlig information. Webbsidor identifierar sig gentemot webbläsaren med ett transportlagersäkerhets (eng. Transport Layer Security, TLS) certifikat. TLS används av HTTPS för att förhandla om och etablera en krypterad anslutning mellan webbsidan och webbläsaren. Dessa certifikat innehåller namnet för domänen som certifikatet är utfärdat för, samt en digital signatur för en betrodd tredje part som går i godo för äktheten av webbsidan som anges i certifikatet. Dessa betrodda tredje parter kallas certifikatutfärdare (eng. Certificate Authorities, CA). Det finns hundratals olika certifikatutfärdare som kan utfärda certifikat och det finns inga restriktioner för vilka domäner dessa kan utfärda certifikat för. Webbläsare har en lista över betrodda certifikatutfärdare, denna lista används av webbläsaren för att validera certifikat som en webbsida identifierar sig med och om certifikatet är utfärdat av en betrodd certifikatutfärdare litar webbläsaren på att webbsidan är den korrekta och etablerar en krypterad anslutning till webbsidan.

Problemet med systemet med certifikatutfärdare är att det är beroende av att varje certifikatutfärdare utför tillräckligt noggranna kontroller av varje person som ansöker om ett certifikat så att denne person verkligen är den som äger domänen eller är auktoriserad att begära ett certifikat för domänen. Det räcker med att en av alla certifikatutfärdare som är betrodda av exempelvis en webbläsare utfärdar ett felaktigt certifikat för en domän för att alla domäner signerade av den certifikatutfärdaren skulle kunna vara komprometterade. Ett sådant certifikat kan användas av en attackerare för att få en bedräglig sida att se mer genuin ut. Exempelvis skulle en webbläsare lita på en falsk Internetbank om denna webbsida presenterar ett fel utfärdat certifikat men som i övrigt är ett giltigt certifikat förutsatt att certifikatutfärdaren är betrodd utav webbläsaren.

För att förbättra säkerheten och metoderna med vilka certifikat valideras släppte Internet Engineering Task Force (IETF) 2012 en ny protokollstandard som kallas domän baserad autentisering av namngivna entiteter (eng. Domain Based Authentication of Named Entities, DANE). DANE ger möjligheten att publicera information om certifikat genom domännamnsystemet (DNS). Domännamnsystemet gör det möjligt för datorer som kommunicerar över ett nätverk som exempelvis Internet att översätta namn till den motsvarande Internet Protokoll (IP) adressen för webbsidan som skall besökas. När datorer kommunicerar med varandra över ett nätverk gör de detta med hjälp av deras unika IP adress. IP adressen representeras av fyra decimaltal mellan 0—255 separerade med punkter, ett exempel på en IP adress är 172.16.254.1. Domännamnsystemet utvecklades för att underlätta för människor att interagera med datorer genom användning av namn istället för att behöva memorera en unik IP adress för varje webbsida eller tjänst de önskar använda.

Domännamnsystemet har några säkerhetsbrister. För det första finns det ingen mekanism för att verifiera att den DNS server som svarar på en DNS fråga verkligen är den korrekta servern som är auktoriserad att svara på DNS frågan. Det andra är att det inte går att veta om svarsmeddelandet har snappats upp och ändrats på vägen mellan avsändaren och mottagaren av en attackerare. För att öka säkerheten för DNS så utvecklades ett tillägg för DNS (eng. Domain Name System Security Extension, DNSSEC). DNSSEC introducerar digitala signaturer genom användandet av en asymmetrisk krypterings metod.

Asymmetrisk kryptering bygger på användandet av två olika krypteringsnycklar, där den ena är en så kallat privat nyckel som hålls hemlig av ägaren och den andra nyckeln är en publik nyckel som görs tillgänglig för alla. Ett meddelande som krypterats med den ena nyckeln kan bara dekrypteras med den andra nyckeln. Detta innebär att en digital signatur som är genererad med hjälp av den privata nyckeln som hålls hemlig av ägaren kan verifieras av vem som helst som har tillgång till den publika nyckeln. Med DNSSEC signeras all information som lagras i Domännamnsystemet och som används för att svara på DNS frågor med den privata nyckeln. Dessa signaturer kan sedan verifieras med hjälp av den publika nyckeln som även den görs tillgänglig via DNS. På detta sätt är det möjligt att säkerställa att den server som svarar på DNS frågor är den som den utger sig för att vara samt att svarsmeddelandena inte har ändrats på vägen mellan avsändaren och mottagaren.

DANE protokollet förlitar sig på DNSSEC för säkerhet. Datorer som ansluter till en webbsida som använder DANE för att publicera certifikatinformation i DNS, kan inte bara verifiera att DNS servern är den korrekta utan även vara säker på att DNS meddelandena inte har ändrats på vägen. Detta medför att IP adressen som tillhandahålls för den efterfrågade tjänsten troligtvis är den korrekta. På samma sätt kan informationen om certifikatet som ges genom DNS med DANE valideras och informationen kan jämföras med det faktiska certifikatet som webbsidan presenterar. Om av någon anledning det faktiska certifikatet inte stämmer överens med informationen som erhållits via DNS, bör webbläsaren avbryta försöket att upprätta en anslutning, då det är möjligt att webbsidan är osäker och potentiellt bedräglig.

Tidigare i denna text har webbsidor och webbläsare används som exempel men DANE är inte begränsat till att användas bara i dessa fall utan kan teoretiskt sett användas med alla tjänster som använder TLS certifikat, dock krävs det att program och tjänster har stöd för DANE. Några exempel på sådana tjänster är e-post och snabbmeddelande (eng. Instant Messaging, IM) tjänster. Det har dock framgått under genomförandet av studien att det fortfarande saknas stöd för DANE i många program, exempelvis finns det inga webbläsare som har inbyggt stöd för DANE ännu. För att kunna dra nytta av DANE i webbläsare måste användaren ladda ned och installera tillägg för webbläsaren som lägger till den funktionaliteten.

Eftersom DANE är en relativt ny standard är antalet studier på området begränsat. Det finns ett fåtal studier som undersöker olika aspekter av DANE så som prestanda, säkerhet och implementationsgrad bland domäner. Denna studie är baserad på en tidigare studie av Zhu m.fl. (2015) som undersökte implementationsgraden av DANE bland .com och .net domäner. Denna studie fokuserar på att kartlägga implementationsgraden av DANE bland svenska .se domäner. Resultat från denna studie kan vara användbart för företag eller organisationer som har planer på att införa DANE för de tjänster de tillhandahåller så som e-post och webbsidor. Andra organisationer som också kan ha nytta av resultatet är de som utvecklar mjukvaruprogram och behöver veta om det finns en efterfrågan av mjukvarulösningar med stöd för DANE protokollet innan några resurser avsätts för utveckling av sådana lösningar.

I denna studie genomfördes en undersökning bland svenska .se domäner med fokus på tre typer av Internet baserade tjänster som har bekräftat stöd för DANE. Dessa tjänster var webbsidor via HTTPS, e-post och direktmeddelandetjänster.

Resultatet visar att implementationsgraden bland de svenska domänerna fortfarande är mycket låg. Endast 79 stycken domäner som använde DANE hittades i undersökningen vilket är mindre än 0,1% av alla svenska domäner. Detta resultat stämmer bra överens med de resultat Zhu m. fl. (2015) rapporterande i sin studie av .com och .net domänerna. Studien visar också att antalet domäner som infört DNSSEC för ökad säkerhet i DNS är 49% av alla svenska domäner vilket motsvarar 686 471 domäner. Detta är ett stort antal domäner som på ett korrekt sätt skulle kunna implementera DANE för

att öka säkerheten för användarna av deras tjänster. Studien visar också att webbsidor med HTTPS är den vanligaste tjänsten som används tillsammans med DANE. Den vanligast typen av certifikat som användes tillsammans med DANE är domänutfärdade certifikat som är ett certifikat som den enskilda domänen utfärdat för sina tjänster utan att blanda in en betrodd tredjeparts certifikatutfärdare.

Resultatet visar att implementationsgraden av DANE ännu är väldigt låg vilket även tidigare studier visat. Detta innebär också att personer som skulle kunna dra nytta av exempelvis webbläsare med inbyggt stöd för DANE är väldigt lågt, vilket troligtvis innebär att mjukvaruutvecklare saknar incitament för att avsätta resurser för att utveckla sådana mjukvarulösningar. För andra tjänster där det redan finns stöd för DANE är implementationsgraden så låg fortfarande att det är få som skulle kunna dra nytta av att en organisation inför DANE för sina tjänster. För att DANE ska kunna få ett större genomslag för exempelvis e-posttjänster behöver större organisationer så som e-post och Internetleverantörer implementera DANE i större skala.

Table of content

1 Introduction

As the demand for secure and encrypted communication over the Internet increases with protocols such as the Hypertext Transfer Protocol Secure (HTTPS) using Secure Socket Layer (SSL) and Transport Layer Security (TLS) certificates to enable secure communication channels. These certificates are signed by trusted third parties called Certificate Authorities (CAs) that bind domain names to these certificates using digital signatures. However, this binding of domain names is done outside of the Domain Name System (DNS) that is authoritative for those domain names. The Internet Engineering Task Force (IETF) wanted to provide an alternative or a way to complement and enhance the security of the Public Key certificate model that was and still is used.

In 2012, the IETF published RFC6698 which contained the specifications for a new protocol: Domain based Authentication of Named Entities (DANE) which allows for TLS certificate information to be published using the DNS infrastructure.

Being a relatively new protocol, there is a limited amount of studies on the various aspects of DANE such as security, performance and implementation levels.

This study aims to establish the current implementation level among second level domains in the Swedish Top Level Domain .se by performing a survey of the authoritative DNS servers for these domains for DANE TLSA resource records.

2 Background

This section aims to provide a basic understanding of the terminology, definitions and technology involved with DNS-Based Authentication of Named Entities (DANE) TLSA protocol as well as provide some insight in to related work.

2.1 DNS

The Domain Name System (DNS) uses a hierarchical tree-like structure, see Figure 1. The top of tree contains the DNS root, often represented by a dot (.). Below the root in the hierarchy are the Top-Level Domains (TLDs) which are divided in two basic categories, Generic Top-Level Domains (gTLDs) and Country Code Top-Level Domains (ccTLDs). Examples of gTLDs are .com, .net and .org whereas ccTLDs are .se, .uk and .us (Aitchison, 2011). Below the TLDs are the Second-Level Domains (SLDs) followed by any number of lower level domains all separated by a dot (.) (Aitchison, 2011).

Each domain is authoritatively governed by an entity that is responsible for administrating that domain, each authoritative entity may delegate authority of a lower level domain to another party. For example, the DNS root is authoritatively administered by the Internet Assigned Numbers Authority (IANA) which have delegated authority of the ccTLDs to various countries around the world. Each delegation of authority is a zone and each zone contains Resource Records (RRs) such as A and AAAA records which bind names to IPv4 and IPv6 addresses respectively (Aitchison, 2011).

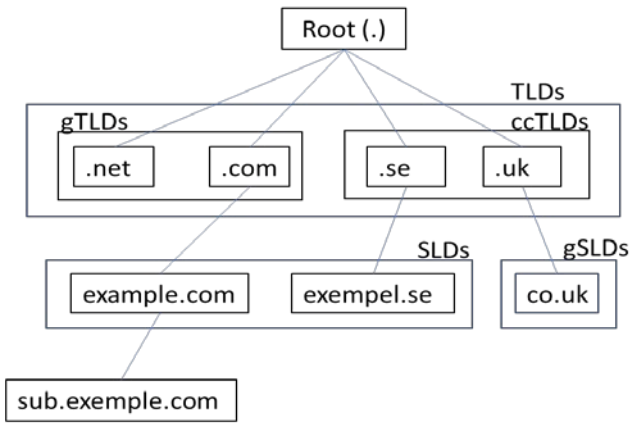

Figure 1 Example showing the DNS hierarchy

The delegation of authority and binding of names to IP-addresses is a fundamental component of the Internet backbone and is what allows users an easy way to access resources on the Internet. If a user wants to visit for example www.example.com. Their local DNS resolver would contact the DNS root servers which in turn would provide a referral to the authoritative name-server for the .com domain. The name-server for .com would similarly provide a referral to the authoritative name-server for the delegated example.com zone. www is in this case a resource located within the example.com zone and thus have an IPv4 RR listed. The name-server would respond to the initial query with the IP address 93.184.216.34 which would then allow the user´s web browser to connect to the www.example.com web site (Aitchison, 2011).

2.2 DNSSEC

DNS is vulnerable to several attacks, one of which is spoofing or cache poisoning. This type of attack allows an attacker to inject information into the DNS server, resulting in unsuspecting users being redirected to a fraudulent web site operated by the attacker. These web sites are usually very convincing replicas of the real web sites such as Internet banks. This would allow the attacker to harvest user data such as login and other personal information (Southam, 2014). These attacks are possible because there is no way to authenticate the origin or the authenticity of the data received when using original DNS (Aitchison, 2011).

To counter this threat, DNS Security Extension (DNSSEC) was introduced. DNSSEC introduces a Public Key Infrastructure (PKI) scheme to DNS using digital signatures of RRs and zones. Public Key cryptography is an asymmetric cryptographic system that uses two keys where plain-text messages encrypted with one key can only be decrypted with the other key. These keys are usually referred to as private and public key where the public key can be made available to anyone while the private key is kept secret. This allows anyone in possession of a public key to receive messages encrypted with the corresponding private key to be sure of the origin of the message as well as its integrity as long as the private key has not been compromised (Aitchison, 2011).

The PKI scheme used with DNSSEC uses digital signatures to authenticate data origin and integrity but not confidentiality e g. the origin and integrity of the DNS replies can be validated but the messages are sent in plain text (Aitchison, 2011).

DNSSEC introduces several new RRs used in DNSSEC enabled zones (Arends et al., 2005):

- DNSKEY – Public key corresponding to the private key used to sign the zone RRs
- RRSIG – Contains the digital signature of a RR signed with the private key
- Next Secure (NSEC) – Provides proof of non-existence (PNE). NSEC provides two forms of PNE, each RR has a corresponding NSEC RR which point to the next valid host in the zone creating a chain of valid hostnames, anything not included in this chain do not exist. The second form contains lists of RR types with the same name as NSEC RR similarly anything not included in the list does not exist.
- Delegation Signer **(**DS) – Contains a hash of the DNSKEY RR. Unlike the DNSKEY RR the DS RR is not stored in the same zone but rather in the signed parent zone to create a chain of trust between zones i e. the DS RR for example.com zone would be stored in .com zone

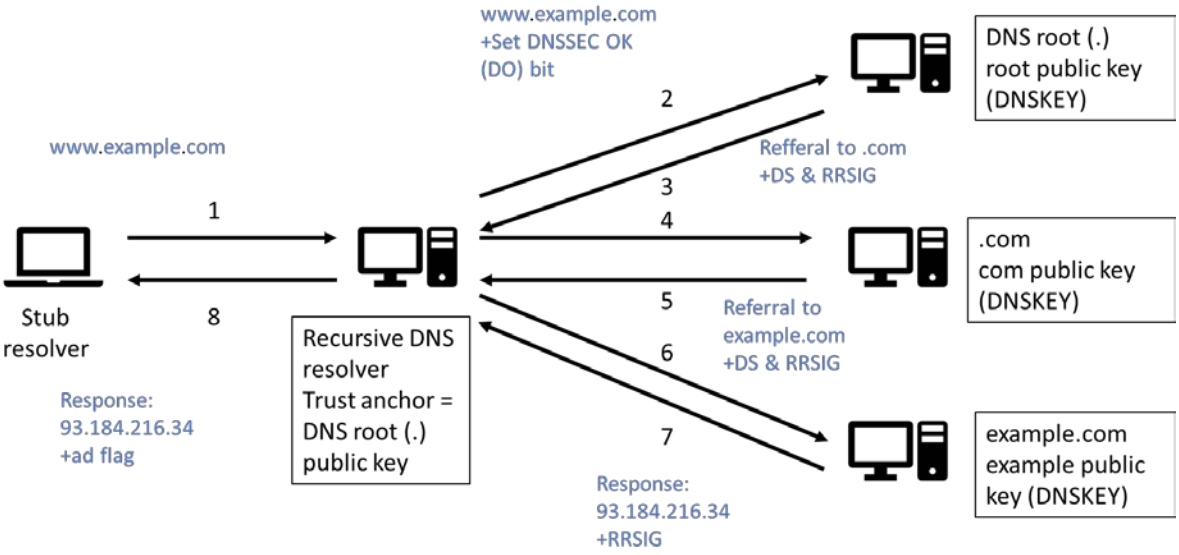

Figure 2 Example of a client making a DNS query for www.example.com using a security aware recursive DNS resolver

Two important concepts to understand are trust anchor and chain of trust. Figure 2 shows the steps involved performing a DNS query for a webserver in the example.com domain using a DNSSEC enabled resolver. The trust anchor is the public key a resolver is configured to trust, in most cases this will be the public key corresponding to the private key used to sign the DNS root at the top of the DNS hierarchy. A chain of trust is formed when there are valid DNSKEY RRs that can be validated using the DS RRs in parent zones all the way to the trust anchor (Aishwarya et al., 2015; Aitchison, 2011; Arends et al., 2005).

2.3 DANE

Secure Sockets Layer (SSL) and its successor Transport Layer Security (TLS) are often used to provide secure communication over the Internet, using channel encryption. TLS uses certificates to bind names and encryption keys. A certificate is composed of a published public key and the name of the service that uses that key. This combination is then digitally signed by another key. For it to be useful to include a key in the certificate the key used to sign the certificate needs to be trusted. This has for a long time been the responsibility of Certification Authorities (CAs). These CAs usually implement strong security measures to protect their secret key and provide their public key to software developers who create TLS client-software. CAs then sign certificates and supply these to servers providing TLS enables services. The TLS clients can use the supplied public keys as trust anchors to validate the certificates presented from these servers (Hoffman & Schlyter, 2012).

DNS-based Authentication of Named Entities (DANE) provides the option to leverage the DNSSEC infrastructure to store, sign keys and certificates used in TLS communication either as a complement to the already existing public CA model or completely separately, allowing domain owners to issue their own certificates without the involvement of a CA (Hoffman & Schlyter, 2012; Zhu et al., 2015).

DANE introduces a new RR type TLSA to DNS and uses the DANE TLSA protocol to retrieve certificate related information relevant to a domain. A TLSA RR contain four RDATA fields:

- Certificate usage field
- The selector field

- The matching type field
- The certificate association field

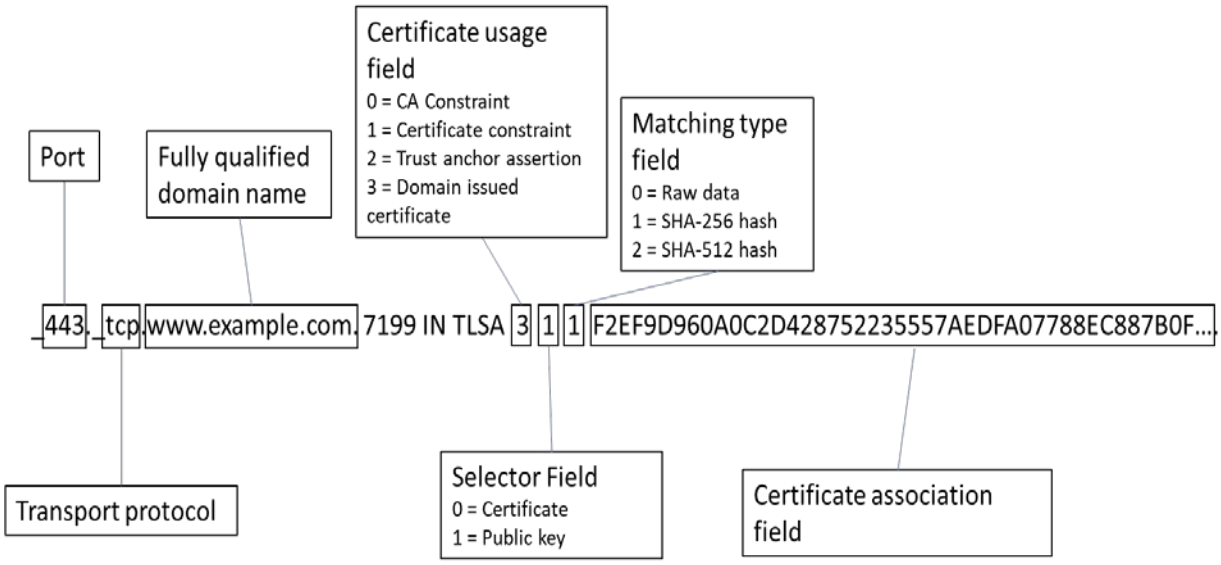

Figure 3 Example of a DANE TLSA resource record

The certificate usage field is one octet long and may have the value 0 – 3. When the value 0 is used, this specifies a CA certificate or the public key of such a certificate that needs to be present in any of the PKIX certification paths for the certificate provided by the server. This usage type is also referred to as a CA constraint as it restricts which CAs can be used to issue certificates for a given service.

A certificate usage field with a value of 1 specifies an end entity certificate or public key of such certificate that needs to match with the certificate provided by the server. This usage type is also referred to as a service certificate constraint as it restricts which end entity certificate can be used by a given service (Hoffman & Schlyter, 2012).

Certificate usage value 2 specifies a certificate or public key that must be used as a trust anchor. This type can be used when a domain issues a certificate under its own CA and that CA is unlikely to be present in the clients list of trust anchors. This type is also referred to as a trust anchor assertion (Aishwarya et al., 2015; Zhu et al., 2015; Hoffman & Schlyter, 2012).

Certificate usage 3 specifies a certificate or public key that must match the certificate provided by the server. The fundamental difference with certificate usage 3 is that in the case of usage 0 – 2 the certificate must in addition to match the TLSA record also pass Public Key Infrastructure for X.509 certificates (PKIX) validation which is not performed in usage case 3. This usage type is also referred to as domain issued certificate as it allows a DNS administrator to issue certificates for a service without involving a third-party CA (Aishwarya et al., 2015; Zhu et al., 2015; Hoffman & Schlyter, 2012).

The selector field specifies which part of the server certificate that will be matched with the association data in the TLSA record. The selector field is one octet long and may have the value 0 or 1, where a value of 0 indicates that the full certificate must match or if the value is 1 only the public key must match (Aishwarya et al., 2015; Zhu et al., 2015; Hoffman & Schlyter, 2012).

The matching type field is a one octet field that may have the values 0 – 2 where 0 is exact match i e. the entire certificate. Matching type 1 indicates a SHA-256 hash needs to be applied and type 2 is a SHA-512 hash (Zhu et al., 2015).

The certificate association data field contains the data that must match as dictated by the selector and matching type fields, i e. the full certificate, a SHA-256 or a SHA-512 hash (Aishwarya et al., 2015).

2.4 Related work

Despite that the DANE protocol is a relatively new protocol there are a few studies that examine different aspects of the DANE protocol as well as contrast it to other proposed methods of authenticating certificates.

Brown and Jenkins (2016) analyse five proposed methods for improving the authentication component of TLS/SSL protected Web services of which DANE is one. The researchers establish several properties by which to compare the different methods such as authentication properties, forensic and privacy properties, usability and implementation properties. Brown and Jenkins (2016) find that DANE offers a very strong domain name authentication. By DANE using DNSSEC to distribute certificate assertions, a client can be guaranteed that these certificate assertions really belong to the domain name in question. The researchers see the DANE TLSA usage 2 – 3 as problematic due to no certification path validation is performed in usage case 3 and for usage case 2 a client is forced to accept the mandated trust anchor even if this is not present in the clients' trust store, additionally Brown and Jenkins claim that usage case 2 – 3 would allow an attacker who can subvert DNSSEC to perform a man in the middle attack.

Osterweil et al. (2014) propose a methodology to quantify the attack surface of networked systems as well as visually represent semantically different components of those systems. To demonstrate their method, they conduct a case study involving the X.509 CA verification system and DANE. The researchers define the attack surface as a measurable set of elements that can be used by an attacker or adversaries. Applying their method to approximately 600 popular web sites using HTTPS they found that the attack surface was two magnitudes larger than it would be with DNSSEC and DNSSEC was one magnitude larger than it would be with DANE.

Aishwarya et al. (2015) examines and analyses the performance of the DANE protocol on the client side as well as presents a tool used for deploying and administrating DANE with name servers in a local network. Their findings show that there is not much of an overhead when using DANE TLSA in terms of bandwidth or CPU usage. When measuring the CPU overhead at the client side they used Mozilla Firefox using a third-party DNSSEC/TLSA validator as there currently is no native support for DANE TLSA in any current web browsers.

A study performed by Zhu et al. (2015) measures the deployment of DANE TLSA in two gTLDs, .com and .net. The researchers actively probed DNSSEC enabled zones for TLSA records for three different protocols, HTTPS, SMTP and XMPP. The researchers only probed DNSSEC enabled zones because TLSA records are only fully trustworthy if integrity can be validated using DNSSEC. Zhu et al. (2015) scanned 130.30M zones and found 485k (0.37%) DNSSEC signed zones of which 443 used TLSA. The researchers also measured growth over time and find that DANE deployment is steadily increasing and appears to show a linear growth trend which could indicate that the population could double in six months.

3 Problem definition

As the Internet grows and the amount of sensitive information that is shared grows the need for secure communication channels increase, and this is handled by web browsers using TLS. TLS requires certificates that bind a certain hostname or service on a specific host to a public key. These certificates are then digitally signed by a CA. There is a large amount of CAs, where each can sign certificates for

any domain and each CA can create subordinate CAs who can sign certificates on behalf of the main CA (Aishwarya et al., 2015).

Many web browsers come configured with a large amount of trusted CA certificates all of which act as trust anchors for clients. If any one of those CAs issues bad certificates either voluntarily or through having their security compromised the whole PKI system may be compromised and lead to an attacker being able to impersonate high value domains such as Internet banks to any web browser that trusts a miss-issued certificate from a compromised CA (Aishwarya et al., 2015).

Using DNSSEC and the DNS root as a single trust anchor instead of the traditional PKI certificate model with its many trusted root CAs would significantly reduce the attack surface and vulnerability of the system (Zhu et al., 2015). DANE TLSA (RFC6698) uses the DNSSEC trust chains to authenticate TLS certificates and can be used to complement the existing public CA PKI model or allow domain administrators to issue their own certificates with certificate usage field type 3 as described in section 2.3 DANE (Hoffman & Schlyter, 2012).

Zhu et al. (2015) examine the DANE TLSA deployment in two TLDs showing that deployment is still in the early stages and propose to measure deployment in zones with a larger set of DNSSEC signed zones. Internet Society (ISOC) (2016) shows that DNSSEC deployment is growing showing that 89% of TLDs have been signed. Verisign labs Secspider (http://secspider.verisignlabs.com/islands.html) shows that the Swedish ccTLD .se has the currently largest set of DNSSEC deployment which would make it suitable for measuring DANE TLSA deployment, additionally the zone file for the .se domain is publicly available which gives a complete list of DNSSEC signed zones suitable for probing for TLSA RRs. As there are no previous studies examining the TLSA deployment within the Swedish domain context this is a good opportunity to do so.

The thesis statement for this study is:

This study aims to establish the deployment level of DANE TLSA among SLDs in the ccTLD .se.

The study will be partially based on the study performed by Zhu et al. (2015) in order to answer the following research questions (RQ):

RQ1: How many Zones have valid DNSSEC signatures that can be validated?

Answering research question #1 would provide insight to how many domains that could provide secure TLSA RRs for services hosted within these domains. A DNSSEC protected domain where the chain of trust cannot be validated could indicate a man in the middle attack and as such any certificates provided through TLSA RRs should not be trusted.

RQ2: How many of the validated DNSSEC zones have deployed TLSA resource records?

Research question #2 is critical in order to fulfil the aim of this study as described in the thesis statement as well as to be able to answer RQ3 and RQ4 in any meaningful way.

RQ3: What is the most frequently deployed service for which TLSA resource records are deployed?

Research question #3 would provide an insight to which services are used in conjunction with DANE TLSA within the Swedish domain context and could be useful for developers who wish to know if there is a demand for implementing support for DANE TLSA in client and server applications that interact with those existing services.

RQ4: What is the most common DANE TLSA certificate use case?

Research question #4 is important as it will show the preferred certificate usage i. g. is TLSA used to enhance the traditional certificate PKI model using third party CA certificates (certificate usage: 0-1) or are these domains issuing their own certificates (certificate usage: 2-3). Answering the research questions would also allow the results to be used for comparison with the results from the study performed by Zhu et al. (2015).

Answering the research questions and the thesis statement would be beneficial to any organization or company operating within the Swedish TLD and are evaluating or considering implementing DANE in their infrastructure but are unsure if there is a demand or user base for such an implementation. Additionally, any company or organization developing software services and applications and are waiting for demand for DNSEC and DANE TLSA support to reach a certain degree may find the result of this study useful.

4 Method strategy

There are many different strategies that can be applied to answer a study's research questions such as surveys, case study, experiments and design and creation. The strategy is the overall approach to answering the research questions (Oates, 2006).

For this study, a survey strategy was selected. A survey approach is appropriate as it focuses on gathering the same kind of data from a large sample population (Oates, 2006). In the context of this study, TLSA RRs from a large group of SLDs within the TLD .se. According to Oates (2006) there are six activities involved with planning and conducting a survey:

- Data requirements
- Data generation method
- Sampling frame
- Sampling technique
- Response rate
- Non-responses

The data requirements dictate what data a researcher wants to generate. The data can be directly related to the research questions or indirectly (Oates, 2006). For this study, the data requirement is:

- TLSA RRs
- DNSSEC validation data

The existence of TLSA RRs within a domain would help answer RQ2: How many of the validated DNSSEC zones have deployed TLSA resource records? A TLSA RR do also contain all the required information to answer RQ3 and RQ4. A security aware DNS resolver that is configured to perform DNSSEC validation will log errors when DNSSEC RRs cannot be validated properly. Such error messages can be used to answer RQ1.

The data gathering method is the method by which the research data is gathered. Surveys often use questionnaires and statistical analysis (Berndtsson et al., 2002). However, several other methods may be used to gather data such as interviews, observations and document based data gathering methods (Oates, 2006). The method selected to generate research data for this study leverages the existing public

DNS infrastructure by interrogating authoritative DNS servers for TLSA RRs for the domains within the selected sample population. Additional data will be collected from logfiles generated from the resolvers used during the data gathering process. This log data will relate to DNSSEC errors and non-connectivity issues. The data gathering method is explained in detail in section 5.

Sampling frame refers to a list or a collection of the whole population that can be included in the survey. From the sampling frame the sample is selected using a sampling technique. The sampling frame must include the whole population that is being studied (Oates, 2006). The selected sampling frame for this study are all registered SLDs contained within the TLD .se zone file. This includes both DNSSEC signed zones and zones that do not use DNSSEC which amounts to 1 406 335 domains as of 2017-04-18.

There are two types of sampling: probability and non-probability sampling techniques. Probability sampling techniques include random, systematic, stratified and cluster sampling. Examples of non-probability sampling techniques are purposive, snowball, self-selection and convenience sampling. The sampling technique used for this study is purposive sampling which is a technique where the researcher picks the sample based on instances that are likely to generate valuable data (Oates, 2006). The sample includes all DNSSEC signed zones with DS RRs listed in the TLD .se zone file. The number of domains included in the sample is 686 471 or 49 % of the total population of SLDs. The reason for excluding non DNSSEC zones is that using DANE TLSA without DNSSEC is considered an error (Zhu et al., 2015).

Response rate and non-responses is more applicable to traditional questionnaire based surveys and relates to the respondent's motivation to respond and the researcher need a plan for how to increase the likelihood of people responding to achieve a sufficient response rate (Oates, 2006). Since this survey is not dependent on human respondents and their motivations this is less of an issue. However, Internet connected services may be unreachable for any number of reason therefore it needs to be mentioned, this issue is further discussed in section 5 Validity.

4.1 Data analysis method

The research questions posed in this study requires a quantitative analysis to be performed to fully answer them. The data generated from survey research is often quantitative in nature therefore survey research in many cases requires a quantitative analysis method to be used (Oates, 2006). The data gathered for this study could be considered a mix of qualitative information as well as quantitative nominal data values. Nominal data values describe categories and does not necessary have an actual numeric value (Oates, 2006). For example, the certificate usage field of the TLSA RR is represented by a number between 0 and 3 but these values represent a method by which a connecting client should validate a TLS certificate rather than an actual numerical value. The TLSA RR also contains qualitative information such as RR type and Fully Qualified Domain Names (FQDN).

The analysis for the study aims to analyse the gathered data for frequency and ratio for occurring data such as how many domains have TLSA RRs for SMTP on port 25 and what are the ratio of values in the certificate usage field among those domains.

4.2 Alternative methods

In this section, some alternative method strategies and data gathering techniques which potentially could be used to investigate the research problem described in section 3. To establish the current implementation level of DANE TLSA among SLD in the TLD .se there are in the author´s view only two applicable method strategies, one being the selected survey method and the other case study research.

A case study is focused on one instance of something that is being examined, this could be a company, organization or single department. For this study, an argument could be made for a case study would be performed on the TLD .se which is one instance of many TLDs. The SLDs are all subdomains to the TLD investigated (Oates, 2006).

A case study tends to be more in-depth than a survey and use several different data gathering methods such as questionnaires, interviews, observations and documents to achieve a greater understanding and a more detailed insight into the problem investigated (Oates, 2006). As the current implementation level of DANE TLSA among the SLDs in the TLD .se is not yet established this method may be better suited for a follow up study as future work.

In terms of alternative data gathering methods, a questionnaire based survey could have been made, this would however drastically reduce the feasible sample size compared to the currently selected method using the public DNS infrastructure. The previous study by Zhu et al. (2015) shows that implementation levels are still very low and there are very few domains that have deployed DANE TLSA which would make it difficult to select a sample that would produce reliable data using a questionnaire based survey.

5 Validity

In this section, potential validity threats for this study will be addressed. It is important to consider and address validity threats as they may affect the validity of the results (Wohlin et al., 2012).

There are several issues that should be addressed for this study. Firstly, reliability of measures refers to the ability to get the same results if study or experiment is repeated. For this study that's not likely to happen due to the constant change in available domains which increase and decrease as new domains are added and old disappear. This threat is not critical because the non-static state of the Internet is normal. However, it is also important to note that any results presented within this report is only a snapshot of the TLSA implementation level as it was on April 20-21, 2017.

Human error is another issue that could affect the results. Doing many DNS queries manually would very likely cause bad queries to be made. To avoid this threat the data collection will be automated using scripts which allows for standardized query construction and ensure that all domains are examined the same way each time.

The script used for data collection could also be a potential threat to validity. To make sure the script collects the intended data reliably it needs to be tested against domains with known working DANE TLSA RRs. The Internet Society (ISOC) (*http://www.internetsociety.org/deploy360/resources/dane-test-sites/*) publishes a list of test domains with known good as well as faulty implementations of DANE TLSA. These sites will be used to test and evaluate the script prior to deployment.

As the data collected is mostly nominal or categorical data it is not possible to do any in-depth statistical analyses to determine significance levels as these tests require mean and standard deviation values to be calculated which according to Oates (2006) makes no sense for nominal data. However, using a large sample size as the one selected for this study which includes all DNSSEC signed SLD in the TLD .se should still provide an accurate insight into the implementation levels of DANE TLSA among Swedish SLDs.

Response rates and non-responses could impact the results. For this study, this means that some DNS servers may be unresponsive due to being offline or configured not to respond to DNS queries. This is not something that can be prevented but the number of unresponsive hosts will be disclosed in the results section. Similarly, there is a risk that the large amount of DNS queries performed during the data

collection could be perceived as suspicious by anomaly based intrusion detection systems or by watchful system administrators and cause a block and disruption of service. In this case, the data collection will be moved and performed at a backup location, in this case the University of Skövdes network where such traffic would appear much less suspicious.

6 Method implementation

In this section, the implementation details for the method described in section 4 will be explained and described in greater detail.

As described in section 4 the selected sample included all DNSSEC signed domains within the TLD .se. To create a list of all domains which could be used for querying each domain´s DNS server for relevant TLSA RRs the zone file for the .se TLD was acquired. The zone file for the .se TLD is published by Internetstiftelsen i Sverige (IIS) and is publicly available at: *https://zonedata.iis.se/*. The zone file used for this study was updated 2017-04-18 05:27:39 and contained records for approximately 1.4 M domains. All domains that had DS RRs were extracted to form a list of DNSSEC signed zones which resulted in a list of 686 471 domains.

The data to gather was limited to TLSA RRs for three different protocols, HTTPS, SMTP and the Extensible Message and Presence Protocol (XMPP). The limitation to these three protocols is due to DANE being a relatively new protocol and the availability of services and applications with DANE support is still limited, however these protocols do have services with DANE support implemented as shown by Zhu et al. (2015). Gathering this data is also necessary to do any form of comparison with the study by Zhu et al. (2015).

Ports that will be used for queries for these protocols is 443 for HTTPS. Ports used for SMTP was 25, 465 and 587. Port 465 have been reassigned to be used for igmpv3lite by the Internet Assigned Numbers Authority (IANA) (*http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.txt*) but is still used for mail services as shown by Zhu et al. (2015). Port 587 is Message Submission Agent (MSA) and is supposed to be used for client submission of mails. For simplicity, these ports will be referred to as SMTP when discussing DANE TLSA for mail services within the context of this study. For queries for XMPP the ports 5222 and 5269 will be used where 5222 is used for client to server connections and 5269 is used for server to server connections.

6.1 DNS resolvers

To collect the required data several security aware recursive DNS resolvers where set up. The resolvers were set up on twelve Virtual Machines (VM) running CentOS7 running on a VMware ESXI Hypervisor. The Hypervisor ran on commodity hardware with a 3.5 GHz quadcore Intel CPU and 24 GB of RAM. The machines were connected through a NETGEAR Prosafe GS110TP switch connected to a ASUS RT-AC66U router. The DNS server software used was the Berkeley Internet Name Domain (BIND) by the Internet Systems Consortium (ISC).

Bind9 was configured as a recursive resolver with DNSSEC and DNSSEC validation turned on. Even though the servers were located behind NAT and not accessible from outside the local network the servers were configured to only answer queries from localhost, additionally the servers were only configured for IPv4 connectivity and as such IPv6 functionality was disabled in the bind configuration files. The VMs were allocated 2 GB of RAM and a 30 GB Virtual Disk drive. The machines were named probe1 through 12 and given an IP address in the range 10.1.1.201—10.1.1.212 with the last two digits

corresponding to their hostname for easy access. The configuration files for Bind9 can be found in Appendix D and E.

6.2 Automating data gathering

As manually doing DNS queries would be prone to human error and highly inefficient for the number of queries required for this study a Perl script was written to automate the task. This decreases the time required for data gathering as well as eliminates human induced errors.

The script was designed to take a list of domains for which to run DNS queries and write any found TLSA RRs as comma separated values (CSV) for easy import into a spreadsheet program for analysis. Any found TLSA RRs would be reported as domain, a number of status flags, port number and FQDN, Time to Live (TTL), RR type fields, Certificate Usage Field, Selector Field, Matching Type Field and finally the certificate association field.

The script relies on the domain information groper (dig) tool to perform all DNS queries. The script was written with functions for each service (HTTPS, SMTP, XMPP) to more easily segment and spread out the workload among multiple DNS resolvers.

For SMTP, the script processes one domain at a time and makes queries for mx records which contain hostnames for SMTP hosts. If any SMTP hosts are found and these point to a host within the same domain or another .se domain the script will query for TLSA RRs on ports 25, 465, 587 for every such host.

For HTTPS, the script queries for TLSA RRs on port 443 for hostnames www and the domain, for example www.example.com and example.com which are two common naming schemes for accessing websites.

For XMPP the script will query for SRV RRs as described in RFC6120 (Saint-Andre, 2011) which result in SRV queries for:

- _xmpp-client._tcp.example.se
- _xmpp-server._tcp.example.se

If any SRV RRs are found queries for TLSA records on ports 5222 and 5269 will be performed. However, if no SRV RRs are found the script will query for TLSA RRs for two common hostnames used for XMPP services:

- Jabber.example.se
- xmpp.example.se

The inclusion of these two hostnames is done in order to be able to compare collected data with the study by Zhu et al. (2015) which includes these names. The source code for the Perl script used is included in Appendix C.

6.3 Data gathering

The data gathering took place between April 20 and 21 2017. To decrease the time requirement for data gathering as well as have some redundancy in case one or more resolvers were to stop responding during the data collection process the workload was divided among the twelve configured resolvers. The resolvers were assigned to one of three groups, each group gathering data on TLSA RRs for one of the three services previously discussed (SMTP, HTTPS and XMPP). The list of DNSSEC signed domains

that were to be checked for TLSA RRs were split into four similarly sized files and each file assigned to one of the resolvers in each group. Table 1 shows resolver groupings and assigned tasks as well as the number of domains each resolver is assigned to query TLSA RRs for.

Table 1. Services and resolver assignment as well as number of domains for each resolver to query for TLSA RRs

Service	Ports	host	IP	Input file	Output file	#Domains
SMTP	25 465 587	Probe1	10.1.1.201	domain-list-part00	smtp-tlsa-part00.csv	180391
		Probe2	10.1.1.202	domain-list-part01	smtp-tlsa-part01.csv	175772
		Probe3	10.1.1.203	domain-list-part02	smtp-tlsa-part02.csv	178971
		Probe4	10.1.1.204	domain-list-part03	smtp-tlsa-part03.csv	151337
HTTPS	443	Probe 5	10.1.1.205	domain-list-part00	https-tlsa-part00.csv	180391
		Probe 6	10.1.1.206	domain-list-part01	https-tlsa-part01.csv	175772
		Probe 7	10.1.1.207	domain-list-part02	https-tlsa-part02.csv	178971
		Probe 8	10.1.1.208	domain-list-part03	https-tlsa-part03.csv	151337
XMPP	5222 5269	Probe 9	10.1.1.209	domain-list-part00	xmpp-tlsa-part00.csv	180391
		Probe 10	10.1.1.210	domain-list-part01	xmpp-tlsa-part01.csv	175772
		Probe 11	10.1.1.211	domain-list-part02	xmpp-tlsa-part02.csv	178971
		Probe 12	10.1.1.212	domain-list-part03	xmpp-tlsa-part03.csv	151337

Once the data collection was completed the output files for each service were concatenated into a larger file which was then imported into Microsoft Excel for analysis.

To gather data on DNSSEC validation failures the log file located in /var/log/messages was parsed for any messages related to DNSSEC validation errors such as:

- No valid RRSIG
- No valid signature found (DS)
- Broken trust chain
- Insecurity proof failed
- Got insecure response; parent indicates it should be secure
- Verify failed due to bad signature

For each such message, the failing domain name was extracted and added to a new file which then had any duplicate entries removed so an accurate count could be made. A similar approach was used to extract domain names for domains with unresponsive DNS servers. To collect data related to unresponsive DNS servers any message containing *servfail* or *refused* error messages had the corresponding domain name extracted to a separate file and had duplicates removed.

6.4 Data analysis

The collected data was imported into a Microsoft Excel spreadsheet for analysis. As most of the collected data was nominal data, the number of statistical tests and calculations that could be performed on the data was limited. For instance, it makes no sense to calculate mean or standard deviation for nominal values that do not have a real numerical value to them (Oates, 2006). Microsoft Excel was used because it is readily available and is a sufficiently powerful tool for the analysis required for this study.

The focus for the data analysis was on calculating frequency of occurrences for all fields in the collected data as well as proportions among groups that could be used for reporting various aspects of the collected data.

As each domain could have multiple TLSA RRs which could generate duplicate entries great care was taken during the analysis by using Microsoft Excel´s build in functions and formulas to ensure each result was correct.

7 Results

In this section, the results related to the research questions are presented. The research questions presented in section 3 were:

RQ1: How many Zones have valid DNSSEC signatures that can be validated?

RQ2: How many of the validated DNSSEC zones have deployed TLSA resource records?

RQ3: What is the most frequently deployed service for which TLSA resource records are deployed?

RQ4: What is the most common DANE TLSA certificate use case?

The first research question relates to the number of DNSSEC signed domains that successfully could be validated and as such host secure TLSA RRs. Table 2 shows domain statistics for SLDs in the TLD .se on April 20-21, 2017.

Table 2. Domain statistics

Total number of SLDs in TLD .se	1 406 335
Number of Non-DNSSEC zones	719 864
Number of DNSSEC Zones	686 471
Number of faulty DNSSEC zones	2 134
Number of domains with Unresponsive DNS	7 896
Number of Validated DNSSEC zones	676 441

There were approximately 1.4 M SLDs within the .se TLD of which ~719k were non-DNSSEC signed domains and ~686k DNSSEC signed domains. Out of the ~686k signed zones there were 2134 domains that failed DNSSEC validation during the data collection process and an additional 7896 domains that had DNS servers that was either offline at the time or configured not to respond to queries. The number of domains that passed DNSSEC validation during data collection was ~676k. Figure 4 shows proportions in percent of validated, unresponsive and zones which failed DNSSEC validation.

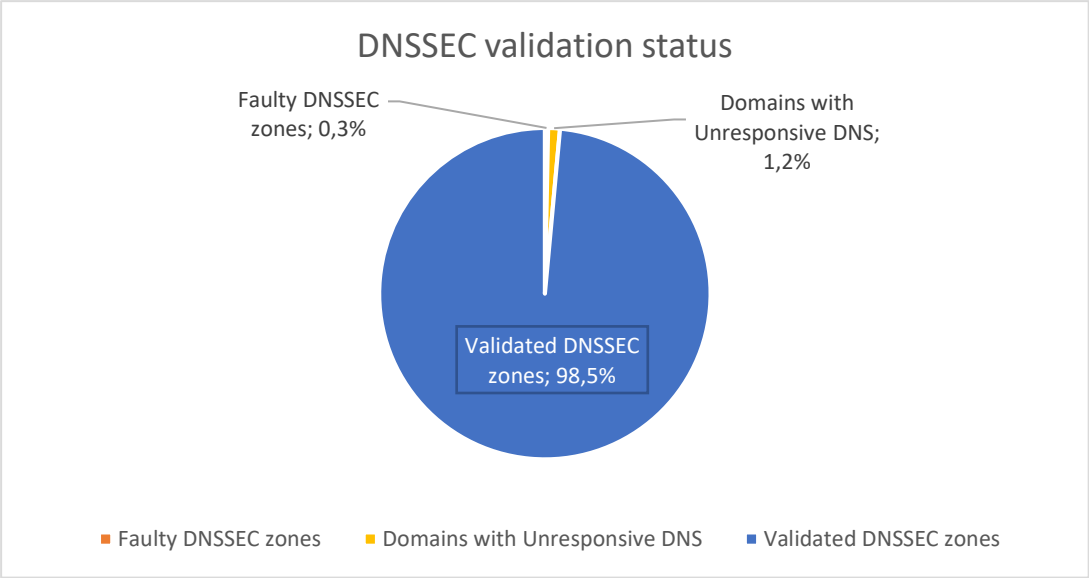

Figure 4. Distribution of validated, faulty and unresponsive DNS domains

The second research question concerns the number of validated DNSSEC domains that have deployed TLSA RRs. Table 3 shows that there were 30 domains with TLSA RRs for SMTP on port 25, 465, 587 and there were 59 domains which had TLSA RRs for HTTPS on port 443. The number of domains using TLSA for XMPP on ports 5222 and 5269 was six. The number if unique domains found to have deployed TLSA RRs was 79 which is approximately 0.01% of the validated DNSSEC Swedish domains. The numbers presented in Table 3 shows more than 79 domains, this is because some domains have deployed TLSA RRs for several services.

Table 3. Domain statistics which had deployed TLSA RRs

Number of domains with TLSA RRs on ports 25, 465 and 587	30
Number of domains with TLSA RRSs on port 443	59
Number of domains with TLSA RRs on ports 5222 and 5269	6

The third research question relates to which service is the most commonly used in conjunction with DANE TLSA. Table 4 shows that the total number of TLSA RRs found was 175 of which 52 were for SMTP. HTTPS had the most deployed TLSA RRs at 108 while services for XMPP only had 15 TLSA RRs deployed.

Table 4. Number of TLSA RRs

Total number of TLSA RRs	175
Number of TLSA RRs for SMTP	52
Number of TLSA RRs for HTTPS	108
Number of TLSA RRs for XMPP	15

Figure 5 shows the distribution of TLSA RRs among the three examined services. HTTPs have 62% of all TLSA RRs. 30% of the TLSA RRs belong to SMTP services and only 9% are used by XMPP services.

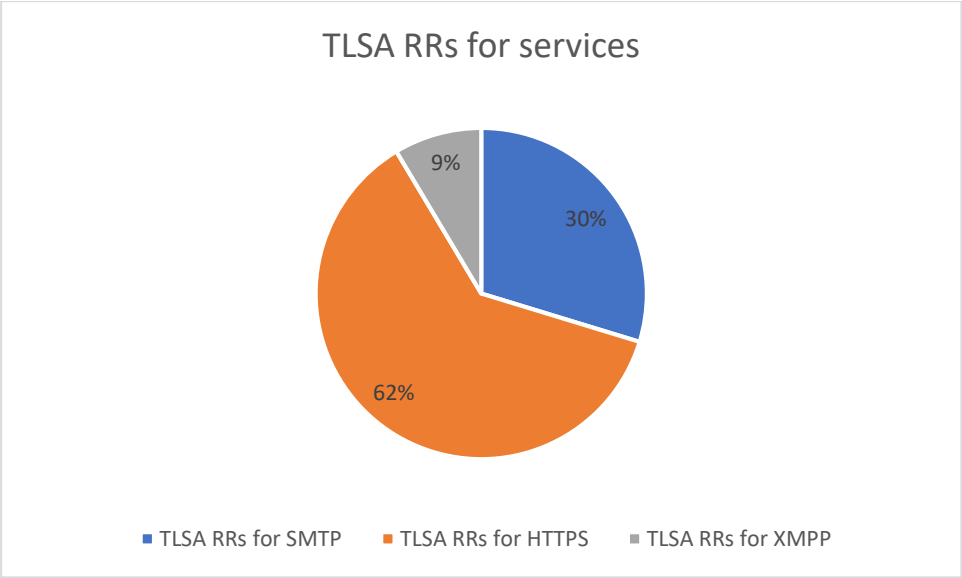

Figure 5. Distribution of TLSA RRs among services

Table 5 shows the number of TLSA RRs per port. HTTPS on port 443 has the largest amount of TLSA RRs and the traditional SMTP port 25 has the second largest group of TLSA RRs. The other two mail ports 465, 587 and the XMPP ports 5222 and 5269 have similar number of TLSA RRs.

Table 5. TLSA RRs numbers broken down by port

Number of TLSA RRs port 25	40
Number of TLSA RRs port 465	7
Number of TLSA RRs port 587	5
Number of TLSA RRs port 443	108
Number of TLSA RRs port 5222	6
Number of TLSA RRs port 5269	9

Figure 6 shows the numbers presented Table 5 of TLSA RRs per port as percentages.

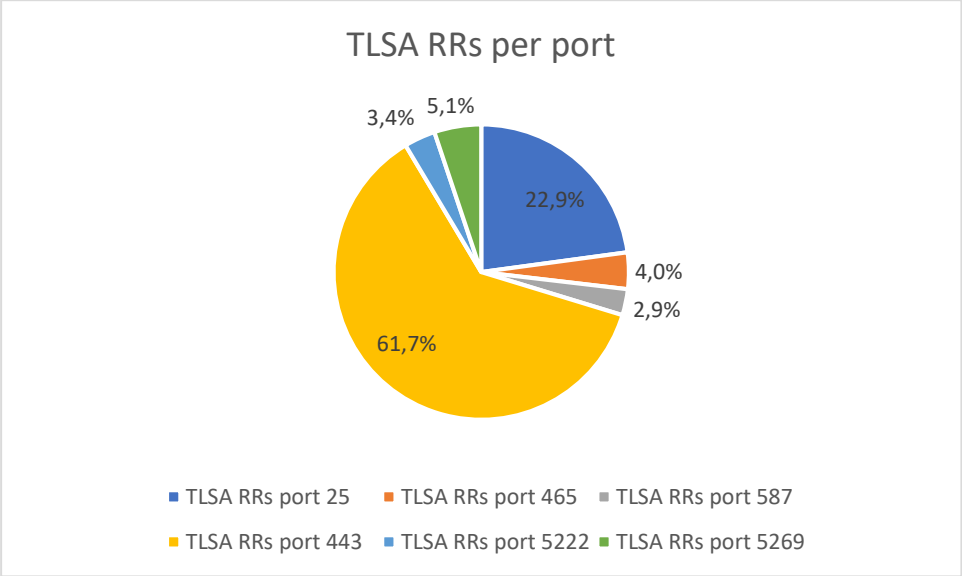

Figure 6. Distribution of TLSA RRs per port

The fourth and final research question relates to the Certificate Usage Field (CUF), specifically which use case is the most common. Table 6 shows the number of TLSA RRs per possible value set in the Certificate Usage Field. For TLSA RRs for SMTP services the most common certificate usage is three which is a domain issued certificate without requiring PKIX validation. For HTTPS services Certificate Usage two is the most common which is trust anchor assertion which specifies which public key or certificate that should be used as a trust anchor.

Table 6. Number of TLSA RRs per Certificate Usage Field option

	Certificate Usage Field set to 0	Certificate Usage Field set to 1	Certificate Usage Field set to 2	Certificate Usage Field set to 3
Options used for TLSA SMTP RRs	0	0	2	50
Options used for TLSA HTTPS RRs	0	5	53	50
Options used for TLSA XMPP RRs	0	0	0	15
Options used for all TLSA RRs	0	5	55	115

HTTPS also have five TLSA RRs with Certificate usage one which is a certificate constraint which certificate must match and pass PKIX validation. All XMPP related TLSA RRs use three as the selected Certificate Usage Field option. Figure 7 shows the distribution of TLSA RRs as percentages for each possible setting for the Certificate Usage Field.

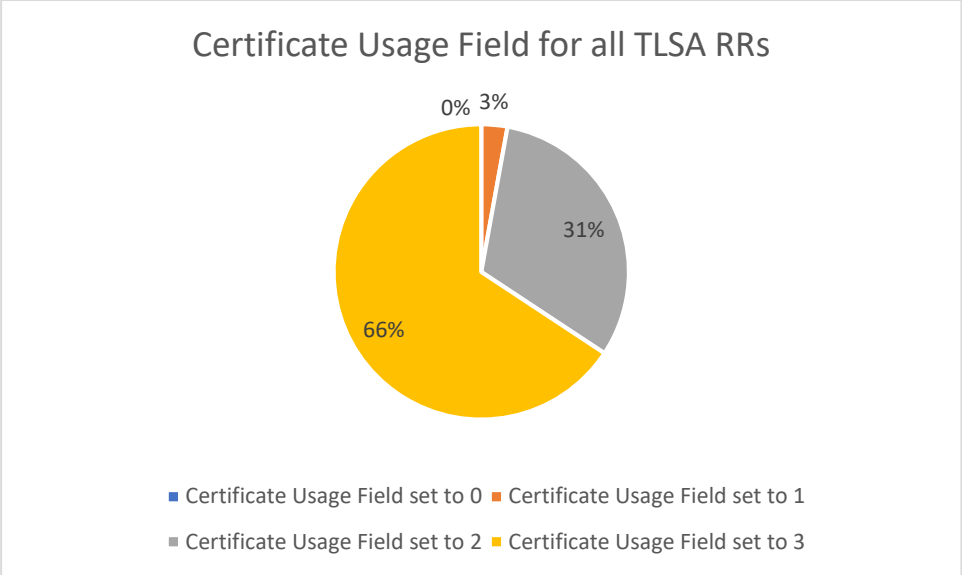

Figure 7. Certificate Usage Field settings for all TLSA RRs

Certificate usage three is the most common usage at 66%. Using certificate usage field setting two is the second largest at 31% and certificate usages one and zero are the least used options at 3% and 0% respectively.

7.1 Comparison to related work

In this section, the results of this study will be compared to those of Zhu et al. (2015) where applicable. Firstly, DNSSEC signed SLDs within the .se TLD was on April 18, 2017 ~686k domains whereas the number of signed domains for .com and .net was ~485k as reported by Zhu et al. (2015).

The number of SLDs with deployed TLSA RRs for TLD .se was 79 and 365 SLDs for .com and .net TLDs. This is approximately 0.012% for the .se SLDs and 0.075% .com and .net SLDs found by Zhu et al. (2015). Both studies find that the deployment level for DANE TLSA is less than 0.1% of DNSSEC signed SLDs.

Table 7 shows a comparison between the study by Zhu et al. (2015) and this study and both studies show that HTTPS (port 443) is the most common service for which DANE TLSA is deployed although HTTPS ratio for .se SLDs is ~20% higher than among SLDs in the .com .net zones.

Table 7. Comparison of proportions of identified TLSA RRs per port for Zhu et al. (2015) and this study

		Zhu et al. (2015)	This study
HTTPS	port 443	39,5%	61,7%
SMTP	port 25	31,5%	22,9%
	port 465	8,7%	4,0%
	port 587	10,5%	2,9%
XMPP	port 5222	4,9%	3,4%
	port 5269	4,9%	5,1%

For Certificate Usage fields Zhu et al. (2015) report that certificate usage three is the most common one with 76% of the TLSA RRs found. This is also the most common certificate usage for this study which show certificate usage three is used with 66% of the found TLSA RRs among .se SLDs.

Overall, the results of this study appear to align with the findings of Zhu et al. (2015) where applicable comparisons can be made.

7.2 Conclusion

This study aimed to establish the deployment level of DANE TLSA among SLDs within the .se TLD, this goal has been achieved along with successfully answering the research questions.

The number of DNSSEC signed domains that were successfully validated was large. At approximately 676k domains this is large group of domains that could deploy DANE TLSA for their provided services to increase security for users of these services. This provides an answer for research question number one.

The study shows that at April 20-21, 2017 the deployment level of DANE TLSA among Swedish SLDs is extremely low with only 79 (~0.01%) out of the available DNSSEC signed zones implementing DANE TLSA RRs in DNS. This provides an answer for research question number two.

The study shows that the most common service deployed in conjunction with DANE TLSA is services using HTTPS over port 443. Although the number of domains implementing DANE TLSA for HTTPS is still very few, only 59 domains have TLSA RRs in DNS for HTTPS. It is possible that the current lack of native support for DANE TLSA in web browsers is holding deployment levels down. This provides an answer for research question number three.

The certificate usage three is used by 66% of domains deploying DANE TLSA RRs and usage two another 31%, and together these cases are used by 97% of the domains deploying DANE TLSA RRs. This finding is interesting as both these use cases do not use the traditional public web PKI model or PKIX validation to validate the certificates. This indicates a reluctance to rely on third party CAs to sign certificates. This provides an answer for research question number four.

8 Discussion and future work

Some ethical aspects that needed consideration during the execution of this study were that it might be possible to identify people, domain holders or administrators and company employees which could potentially cause these people harm e.g. to have their employment terminated. The reasoning is that in conducting this study, several domains that have issues with configuration as well as poorly maintained zones were encountered. If this information was published in the report it may have a negative impact on these people, and for this reason no information regarding domains or information related to any people related to the domains included in this study is published or included in this report.

Another aspect to consider is that in performing the data collection in the way described in section 6 causes abnormal additional traffic that would otherwise not exist. However, the impact on any individual name server was deemed acceptable as the number of queries to any specific domain was not very large. The total number of IPv4 DNS queries generated during the data collection was approximately 13 M.

The results of this study show that the deployment levels of DANE among Swedish SLDs are very low which is similar to the result of the previous study by Zhu et al. (2015). This indicates that the demand for software solutions with DANE support is limited. For example, there are currently no web browsers available that have native support for DANE and due to the low number of web servers using HTTPS with DANE, software developers lack incentives to devote resources towards the development of such applications. Another type of organization that could benefit from the results of this study are organizations that are considering deploying DANE for their own services but are unsure whether there is a sufficiently large deployment of DANE supported services in place to benefit from such a deployment. Current DANE deployment levels among Swedish SLDs suggests that the benefits of deploying DANE with services such as e-email would be limited as there are very few instances where DANE would be used to establish secure connection between hosts.

This study provided a snapshot of DANE TLSA deployment levels among Swedish SLDs using a survey method strategy leveraging the public DNS infrastructure to gather the required research data. One drawback with the study is that it only shows a snapshot of the deployment level as it was on April 20-21, 2017 when the data was collected. The study is only capable to show what the current deployment level is but not why it is as low as it turned out to be.

Another aspect that could have been investigated would be what types of organizations have deployed DANE TLSA. There is also a lack of information regarding which applications and services have existing support for DANE TLSA. Furthermore, it is not known what the awareness of DANE is among domain holders and the IT community at large. Based on this discussion the following topics could be examined for future work or extension to this study:

- Examine the awareness of DANE among people in the IT business
- Examine and collect information related to software, both services and client applications with existing or planned support for DANE
- Investigate the types of organizations that have deployed DANE TLSA. This could be combined with a deeper investigation into these organizations´ motivations for deploying DANE
- Measure the deployment level over a longer time period to see how the deployment level develops

References

Aishwarya, C., Raghuram, M. A., Hosmani, S., Sannidhan, M. S., Rajendran, B., Chandrasekaran, K., & Bindhumadhava, K. (2015). DANE: An inbuilt security extension. In *2015 International Conference on Green Computing and Internet of Things* (ICGCIoT). Noida, 2015, (pp. 1571-1576). doi: 10.1109/ICGCIoT.2015.7380717

Arends, R., Austein, R., Larson, M., Massey, D., & Rose, S. (2005). *RFC3034: Resource Records for the DNS Security Extensions*. Retrieved from: https://tools.ietf.org/pdf/rfc3044.pdf

Aitchison, R. (2011). *Pro DNS and BIND 10*. Berkeley, CA: Apress.

Berndtsson, M., Hansson, J., Olsson, B., & Lundell, B. (2002). *Thesis projects: A guide for students in computer science and information systems*. (2nd Ed) London: Springer

Brown, C. W., & Jenkins, M. (2016). Analyzing proposals for improving authentication on the TLS-/SSL-protected Web. International Journal of Information Security, 15(6), 621-635. doi:10.1007/s10207-016-0316-2

Hoffman, P. & Schlyter, J. (2012). *RFC6698: The DNS-Based Authentication of Named Entities (DANE), Transport Layer Security (TLS) Protocol: TLSA.* Internet Engineering Task Force (IETF). Retrieved from: https://tools.ietf.org/pdf/rfc6698.pdf

Internet Society, (2016). State of DNSSEC deployment 2016. (Report: 2016). Retrieved from https://www.internetsociety.org/doc/state-dnssec-deployment-2016

Oates, B. J. (2006). *Researching information systems and computing*. SAGE Publications, London; Thousand Oaks, Calif.

Osterweil, E., McPherson, D., & Zhang, L. (2014, 21-24 Oct. 2014). The Shape and Size of Threats: Defining a Networked System's Attack Surface. Paper presented at the 2014 IEEE 22nd International Conference on Network Protocols. doi: 10.1109/ICNP.2014.101

Saint-Andre, P. (2011). *RFC6120: Extensible Messaging and Presence Protocol (XMPP): Core.* Internet Engineering Task Force (IETF). Retrieved from: https://tools.ietf.org/html/rfc6120

Southam, M. (2014). DNSSEC: What it is and why it matters. *Network Security, 2014*(5), 12-15. doi:10.1016/S1353-4858(14)70050-9

Wohlin, C., Runeson, P., Höst, M., Ohlsson, M. A., Regnell, B. & Wesslén, A. (2012). *Experimentation in software engineering*. New York: Springer

Zhu, L., Wessels, D., Mankin, A., & Heidemann, J. (2015). Measuring DANE TLSA Deployment. In M. Steiner, P. Barlet-Ros, & O. Bonaventure (Eds.), *Traffic Monitoring and Analysis: 7th International Workshop, TMA 2015*, Barcelona, Spain, April 21-24, 2015. Proceedings (pp. 219-232). Cham: Springer International Publishing. DOI: 10.1007/978-3-319-17172-2_15

Appendix A - Original time plan

Week	Dates 2017	Activities	Milestones/Deadlines	Comment
8	20/1 - 26/2		21/2 Problem formulation and backgroud report 23/2 Problem defintion and background slides	21/2 only partial submission made
9	27/2 - 5/3	Participate in seminars all week		
10	6/3 - 12/3	Select method Evaluate validity threats plan for data collection method		
11	13/3 - 19/3	Design scripts for data collection Test scripts an evaluate		
12	20/3 - 26/3	Start collection data Write report method and validity		
13	27/3 - 2/4	Begin analysing initial data		
14	3/4 - 9/4			
15	10/4 - 16/4	prepare report		
16	17/4 - 23/4	Finalize report and slides for seminars		
17	24/4 - 30/4		25/4 Method, realization, and analysis report 27/4 Method, relization, and analysis slides	
18	1/5 - 7/5	Finalizing report		
19	8/5 - 14/5	Finalizing report		
20	15/5 - 21/5	Finalizing report and slides		
21	22/5 - 28/5		23/5 Final report 25/5 Final report slides	
22	29/5 - 4/6			
23	5/6 - 11/6		11/6 "Final" Final report submission	

Appendix B - Updated time plan 2017-04-24

Week	Dates 2017	Activities	Milestones/Deadlines	Comment
8	20/1 - 26/2		21/2 Problem formulation and backgroud report 23/2 Problem defintion and background slides	21/2 only partial submission made
9	27/2 - 5/3	Participate in seminars all week		
10	6/3 - 12/3			
11	13/3 - 19/3			
12	20/3 - 26/3			
13	27/3 - 2/4			
14	3/4 - 9/4			
15	10/4 - 16/4			
16	17/4 - 23/4	Select method Evaluate validity threats plan for data collection method Design scripts for data collection Test scripts an evaluate Start collection data Begin analysing initial data Write report method and validity		Continues in to week 17
17	24/4 - 30/4	prepare report Finalize report and slides for seminars	25/4 Method, realization, and analysis report 27/4 Method, relization, and analysis slides	
18	1/5 - 7/5	Finalize report and slides for seminars		
19	8/5 - 14/5	Finalizing report		
20	15/5 - 21/5	Finalizing report and slides		
21	22/5 - 28/5		23/5 Final report 25/5 Final report slides	
22	29/5 - 4/6			
23	5/6 - 11/6		11/6 "Final" Final report submission	

Appendix C - Source code for data gathering script

```
#!/usr/bin/perl
################ METADATA ###################################################################################
# NAME: Rikard Sandelin
# Email: a14riksa@student.his.se
# COURSE: IT604G, Bachelor Degree Project in Information Technology with a Specialisation in Network and System Administration G2E
# Purpose of script: query all domain for DANE TLSA records for SMTP|HTTPS|XMPP hosts
# DATE OF LAST CHANGE: 2017-04-20
#############################################################################################################

use warnings;
use strict;

#Check arguments given on commandline
my $num_args = $#ARGV + 1;
if ($num_args != 3) {
      print "\nUsage: ./tlsa-smtp.pl [smtp|https|xmpp] /path/to/inputfilename /path/to/outputfilename\n";
      exit;
      };

my $testtype = $ARGV[0];
my $inputfile = $ARGV[1];
my $outputfile = $ARGV[2];
my $counter = 0;

#queries_to_run ();
#Evaluates users choice of what protocol/service to check for DANE TLSA RRs
sub queries_to_run {
if ($testtype !~ /^smtp$|^https$|^xmpp$/){
            print "\nUsage: ./tlsa-smtp.pl [smtp|https|xmpp| /path/to/inputfile_name /path/to/outputfile_name\n";
            exit;
      }
      elsif ($testtype =~ /^smtp$/){
            dig_mx_records ();
      }
      elsif ($testtype =~ /^https$/){
            dig_https_tlsa ();
      }
```

```
        else {
        dig_srv_XMPP_records ();
        }
}

#dig_mx_records ();
#open a filehandle on inputfile and read each line (domain)
#For each domain, query for mx records and store results in array
#for each MX RR in array query for TLSA records if MX RR points to same domain or other .se domain
sub dig_mx_records {
        my @mx_records;
        open (my $domains, '<', $inputfile) || die "cannot open $inputfile ($!)";
        while (my $domain = <$domains>){
                chomp $domain;
                open (my $digmxcmd, "dig $domain mx +short|") || die "cannot open process ($!)";
                while (my $line = <$digmxcmd>) {
                        if ($line =~ /([a-z0-9-]{1,}\.[a-z0-9]{1,}\.[a-z]{2,3})/) {
                                my $mxrecord = $1;
                                push @mx_records, $mxrecord;
                        }
                }
                close $digmxcmd || die "cannot close $digmxcmd ($!)";
                foreach my $mx (@mx_records) {
                        #check if MX record contains domain name or .se
                        #if yes then pass to sub dig_smtp_tlsa ()
                        if (index ($mx, "$domain" || ".se") !=-1) {
                                dig_smtp_tlsa ($mx, $domain)
                        }
                }
                #clear array before next iteration
                @mx_records = ();
                #$counter and print only for activity check (Script still progressing)
                $counter++;
                print "$counter\n";
        }
        #close filehandle on inputfile
        close $domains || die "cannot close $domains ($!)";
}

```

```
#dig_smtp_tlsa ($mx-record, $domain-name);
#Queries for TLSA RRs for every MX RR passed from dig_mx_records () for ports 25, 465 and 587
#if TLSA RRs found print flags and RRs to outputfilename as comma separated values
sub dig_smtp_tlsa {
      my $mx_record = shift @_;
      my $domain = shift @_;
      #ports that may be used for TLSA RRs
      my @ports = ( 25, 465, 587 );
      foreach my $port (@ports){
            my @tlsaresult = ();
            open (my $digtlsacmd, "dig _$port._tcp.$mx_record tlsa +dnssec|") || die "cannot open process ($!)";
            while (my $line = <$digtlsacmd>) {
                  my $flags;
                  if ($line =~ /^;;\sflags:\s(q?r?\sr?d?;?r?d?;?\sr?a?;?\sa?d?;?c?d?;?);/){
                        $flags = $1;
                        @tlsaresult = split /\s+/, $flags, 5;
                  }
                  my  $tlsarecord;
                  if ($line =~ /^(_$port\._tcp.$mx_record\.\s+\d+\s+IN\s+TLSA\s+[0-3]\s[0-1]\s[0-2]\s[a-fA-F0-9\s]+)$/){
                        $tlsarecord = $1;
                        #add to array and split matched TLSA record on whitespace but no more than 8 fields do keep Certificate Association Field
intact
                        push @tlsaresult, split /\s+/, $tlsarecord, 8;
                        #add domain name to first element of array
                        unshift @tlsaresult, $domain;
                        #join contents of @array as a string joinded by ;
                        my $outputtext = join (";", @tlsaresult);
                        #remove trailing \n after Certificate Association Field
                        chomp $outputtext;
                        #open filehandle and print to outputfilename
                        open (my $output, '>>', $outputfile) || die "cannot open $outputfile ($!)";
                        print $output "$outputtext\n";
                        close $output || die "cannot close $outputfile ($!)";
                        @tlsaresult = ()
                  }
            }
            #close filehandle for outputfilename
            close $digtlsacmd || die "cannot close $digtlsacmd ($!)";
      }
```

```
}

#dig_https_tlsa ();
#for every domain in inputfile query for TLSA RRs for $domain and www.$domain
#prints results to outputfilename as comma separated values
sub dig_https_tlsa {
      open (my $domains, '<', $inputfile) || die "cannot open $inputfile ($!)";
      while (my $domain = <$domains>){
            chomp $domain;
            my $persflags = "null";
            my @dig_queries = ("_443._tcp.$domain","_443._tcp.www.$domain");
            foreach my $query (@dig_queries){
                  my @tlsaresult;
                  open (my $dighttpstlsa, "dig $query tlsa +dnssec|") || die "cannot open process ($!)";
                  while (my $line = <$dighttpstlsa>){
                        if ($line =~ /^;;\sflags:\s(q?r?\sr?d?;?r?d?;?\sr?a?;?\sa?d?;?c?d?;?);/){
                              my $flags = $1;
                              $persflags = $flags;
                        }
                        my  $tlsarecord;
                        if ($line =~ /^($query\.\s+\d+\s+IN\s+TLSA\s+[0-3]\s[0-1]\s[0-2]\s[a-fA-F0-9\s]+)$/){
                              $tlsarecord = $1;
                              @tlsaresult = split /\s+/, $persflags, 5;
                              #add to array and split matched TLSA record on whitespace but no more than 8 fields to keep Certificate Association
Field intact
                              push @tlsaresult, split /\s+/, $tlsarecord, 8;
                              unshift @tlsaresult, $domain;
                              #join contents of @array as a string joinded by ;
                              my $outputtext = join (";", @tlsaresult);
                              #remove trailing \n after Certificate Association Field
                              chomp $outputtext;
                              #open filehandle and print to outputfilename
                              open (my $output, '>>', $outputfile) || die "cannot open $outputfile ($!)";
                              print $output "$outputtext\n";
                              close $output || die "cannot close $outputfile ($!)";
                              @tlsaresult = ()
                        }
                  }
                  #close filehandle on dig process
```

```
                  close $dighttpstlsa || die "cannot close $dighttpstlsa ($!)";
            }
            #$counter and print only for activity check (Script still progressing)
            $counter++;
            print "$counter\n";
      }
      #close filehandle on inputfile
      close $domains || die "cannot close $domains ($!)";
}

#dig_srv_XMPP_records ();
# for all domains in inputfile check for SRV records (rfc6120:3.2.1)
# If no SRV records found try some common XMPP names on ports 5222(client) and 5269(server)
sub dig_srv_XMPP_records {
      open (my $domains, '<', $inputfile) || die "cannot open $inputfile ($!)";
      while (my $domain = <$domains>){
            chomp $domain;
            #construct queries according to rfc6120
            my @dig_queries = ("_xmpp-client._tcp.$domain","_xmpp-server._tcp.$domain");
            my @srv_results = ();
            foreach my $query (@dig_queries){
                  #open filehandle to dig process and run queries
                  open (my $dig_xmpp_srv, "dig $query srv +short|") || die "cannot open process ($!)";
                  while (my $line = <$dig_xmpp_srv>){
                        #if dig result in a match, add to @array
                        if ($line =~ /^\d+\s+\d+\s+(\d+)\s+([a-z0-9]+\.$domain)\.$/){
                              my ( $port, $hostname ) = ( $1, $2 );
                              push @srv_results, join ("", "_$port._tcp.$hostname");
                        }
                  }
                  #close dig process
                  close $dig_xmpp_srv || die "cannot close $dig_xmpp_srv ($!)";
            }
            #if SRV RRs found pass to dig_xmpp_srv($domain-name, @array);
            if (my $num_srv_rr = (scalar(@srv_results)) > 0 ){
                  dig_xmpp_tlsa ($domain, @srv_results);
            }
            #else try common xmpp names
            elsif ($num_srv_rr = (scalar(@srv_results)) == 0 ){
```

```
                  push @srv_results, ("_5222._tcp.jabber.$domain", "_5222._tcp.xmpp.$domain", "_5269._tcp.jabber.$domain",
"_5269._tcp.xmpp.$domain");
                  dig_xmpp_tlsa ($domain, @srv_results);
            }
            @srv_results = ();
            #$counter and print only for activity check (Script still progressing)
            $counter++;
            print "$counter\n";
      }
      #close filehandle on inputfile
      close $domains || die "cannot close $domains ($!)";
}

#dig_xmpp_tlsa ($domain-name, @SRV_array)
#for each SRV RR found in dig_srv_XMPP_records(); query for TLSA RRs
#if TLSA RRs found, print to outputfilename as comma separated values
sub dig_xmpp_tlsa {
      my $domain = shift @_;
      my @srv_results = @_;
      my $persflags = "null";
      foreach my $result (@srv_results){
            my @tlsaresult =();
            #open dig process
            open (my $digtlsacmd, "dig $result tlsa +dnssec|") || die "cannot open process ($!)";
            while (my $line = <$digtlsacmd> ){
                  #match flags for queries
                  if ($line =~ /^;;\sflags:\s(q?r?\sr?d?;?r?d?;?\sr?a?;?\sa?d?;?c?d?;?);/){
                        my $flags = $1;
                        $persflags = $flags;
                  }
                  my  $tlsarecord;
                  #if line matches a TLSA record add flags, TLSA record and domain name
                  if ($line =~ /^([a-z0-9_\s\.-]+IN\s+TLSA\s+[0-3]\s[0-1]\s[0-2]\s[a-fA-F0-9\s]+)$/){
                        $tlsarecord = $1;
                        @tlsaresult = split /\s+/, $persflags, 5;
                        #split matched TLSA record on whitespace but no more than 8 fields to keep Certificate Association Field intact
                        push @tlsaresult, split /\s+/, $tlsarecord, 8;
                        unshift @tlsaresult, $domain;
                        my $outputtext = join (";", @tlsaresult);
```

```
                        chomp $outputtext;
                        #open filehandle on outputfilename and print results
                        open (my $output, '>>', $outputfile) || die "cannot open $outputfile ($!)";
                        print $output "$outputtext\n";
                        close $output || die "cannot close $outputfile ($!)";
                        @tlsaresult = ()
                    }
            }
            #close dig process
            close $digtlsacmd || die "cannot close $digtlsacmd ($!)";
      }
}
queries_to_run ();
```

Appendix D - Resolver configuration: named.conf

```
// named.conf
//
// Provided by Red Hat bind package to configure the ISC BIND named(8) DNS
// server as a caching only nameserver (as a localhost DNS resolver only).
//
// See /usr/share/doc/bind*/sample/ for example named configuration files.
//
// See the BIND Administrator's Reference Manual (ARM) for details about the
// configuration located in /usr/share/doc/bind-{version}/Bv9ARM.html

options {
      listen-on port 53 { 127.0.0.1; };
//    IPv6 disabled 2017-04-18
//    listen-on-v6 port 53 { ::1; };
      directory   "/var/named";
      dump-file   "/var/named/data/cache_dump.db";
      statistics-file "/var/named/data/named_stats.txt";
      memstatistics-file "/var/named/data/named_mem_stats.txt";
      allow-query     { localhost; };
      //added recursion to default configuration 2017-04-18
      allow-recursion { localhost; };
      /*
       - If you are building an AUTHORITATIVE DNS server, do NOT enable recursion.
       - If you are building a RECURSIVE (caching) DNS server, you need to enable
         recursion.
       - If your recursive DNS server has a public IP address, you MUST enable access
         control to limit queries to your legitimate users. Failing to do so will
         cause your server to become part of large scale DNS amplification
         attacks. Implementing BCP38 within your network would greatly
         reduce such attack surface
      */
      recursion yes;
      dnssec-enable yes;
      dnssec-validation yes;
      //dnssec lookaside auto added 2017-04-18
      dnssec-lookaside auto;
      /* Path to ISC DLV key */
      bindkeys-file "/etc/named.iscdlv.key";

      managed-keys-directory "/var/named/dynamic";

      pid-file "/run/named/named.pid";
      session-keyfile "/run/named/session.key";
};

logging {
        channel default_debug {
                file "data/named.run";
                severity dynamic;
        };
};

zone "." IN {
      type hint;
      file "named.ca";
};

include "/etc/named.rfc1912.zones";
include "/etc/named.root.key";
```

Appendix E - Resolver configuration: named options file

```
# BIND named process options
# ~~~~~~~~~~~~~~~~~~~~~~~~~~
#
# OPTIONS="whatever"     --  These additional options will be passed to named
#                            at startup. Don't add -t here, enable proper
#                            -chroot.service unit file.
#
# DISABLE_ZONE_CHECKING  --  By default, service file calls named-checkzone
#                            utility for every zone to ensure all zones are
#                            valid before named starts. If you set this option
#                            to 'yes' then service file doesn't perform those
#                            checks.
OPTIONS="-4"
```