UNIVERSIDADE DO ESTADO DE SANTA CATARINA – UDESC
CENTRO DE CIÊNCIAS TECNOLÓGICAS – CCT
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO – BCC

MATHEUS RAMBO DA ROZA

ANÁLISE DO ECOSSISTEMA DANE PARA E-MAIL NA INTERNET BRASILEIRA

JOINVILLE
2021

MATHEUS RAMBO DA ROZA

ANÁLISE DO ECOSSISTEMA DANE PARA E-MAIL NA INTERNET BRASILEIRA

Trabalho de conclusão de curso submetido à Universidade do Estado de Santa Catarina como parte dos requisitos para a obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Rafael Rodrigues Obelheiro

JOINVILLE
2021

Para gerar a ficha catalográfica de teses e
dissertações acessar o link:
https://www.udesc.br/bu/manuais/ficha

Rambo da Roza, Matheus
Análise do Ecossistema DANE para E-mail na Internet Brasileira / Matheus Rambo da Roza. - Joinville, 2021.
38 p. : il. ; 30 cm.

Orientador: Rafael Rodrigues Obelheiro.
.
TCC (Bacharelado) - Universidade do Estado de Santa Catarina, Centro de Ciências Tecnológicas, Bacharelado em Ciência da Computação, Joinville, 2021.

1. Palavra-chave. 2. Palavra-chave. 3. Palavra-chave. 4. Palavra-chave. 5. Palavra-chave. I. Rodrigues Obelheiro, Rafael. II. , . III. Universidade do Estado de Santa Catarina, Centro de Ciências Tecnológicas, Bacharelado em Ciência da Computação. IV. Título.

“Cada sonho que você deixa pra trás, é um
pedaço do seu futuro que deixa de existir.”
(Steve Jobs)

RESUMO

O SMTP (Simple Mail Transfer Protocol), o protocolo usado para transporte de correio eletrônico na Internet, não usa criptografia nativa, e por isso não garante confidencialidade, integridade ou autenticação do tráfego. Mecanismos para garantir a segurança do SMTP foram propostos, sendo um dos principais o STARTTLS, que permite o uso de canais criptografados TLS (Transport Layer Security) em transações SMTP. O STARTTLS exige que clientes e servidores SMTP tenham certificados digitais, usualmente emitidos por autoridades certificadoras (ACs). O DANE (DNS-based Authentication of Named Entities) é um padrão da Internet que permite que certificados TLS sejam autenticados usando registros TLSA publicados no DNS pelos donos desses certificados, dispensando a necessidade de autoridades certificadoras. O principal caso de uso do DANE atualmente está na validação dos certificados TLS usados para proteção do tráfego SMTP. Este trabalho propõe realizar um estudo de medições sobre o uso de DANE para proteção do serviço de email na Internet brasileira, identificando a adoção do DANE em domínios brasileiros e analisando se o padrão está sendo usado corretamente. Para isso, pretende-se replicar a metodologia proposta por um trabalho anterior, que realizou um estudo análogo com os domínios genéricos `.com`, `.net` e `.org`, e com os domínios nacionais `.nl` e `.se`.

Palavras-chave: Protocolos de segurança, DANE, Transport Layer Security, Autoridade de certificação, SMTP.

ABSTRACT

SMTP (Simple Mail Transfer Protocol), the protocol used to transport electronic mail on the Internet, does not use native encryption, and therefore does not guarantee traffic confidentiality, integrity, or authentication. Mechanisms to guarantee SMTP security have been proposed, one of the main ones being STARTTLS, which allows the use of TLS (Transport Layer Security) encrypted channels in SMTP transactions. STARTTLS requires SMTP clients and servers to have digital certificates, usually issued by certificate authorities (CAs). DANE (DNS-based Authentication of Named Entities) is an Internet standard that allows TLS certificates to be authenticated using TLSA records published in the DNS by the holders of these certificates, eliminating the need for certification authorities. DANE's main use case today is in the validation of TLS certificates used for SMTP traffic protection. This work proposes to carry out a study of measurements on the use of DANE to protect the email service on the Brazilian Internet, identifying the adoption of DANE in Brazilian domains and analyzing whether the standard is being used correctly. For this, we intend to replicate the methodology proposed by a previous work, which carried out an analogous study with the generic domains `.com`, `.net` and `.org`, and with the national domains `.nl` and `.se`.

Keywords: Security Protocols, DANE, Transport Layer Security, Certification Authority, SMTP.

SUMÁRIO

1	**INTRODUÇÃO**	**7**
1.1	JUSTIFICATIVA	**8**
1.2	OBJETIVOS	**8**
1.2.1	**Objetivo Geral**	**8**
1.2.2	**Objetivos Específicos**	**9**
1.3	METODOLOGIA	**9**
1.4	ORGANIZAÇÃO DO TEXTO	**9**
2	**FUNDAMENTAÇÃO TEÓRICA**	**10**
2.1	DOMAIN NAME SYSTEM (DNS)	**10**
2.1.1	**O que é DNS**	**10**
2.1.2	**Espaço de nomes do DNS**	**11**
2.1.3	**Resolução de nomes**	**14**
2.1.4	**Formato de mensagens DNS**	**15**
2.2	DNSSEC	**17**
2.3	DANE	**20**
2.4	SMTP E STARTTLS	**22**
2.4.1	**Modelo de processamento de email na Internet**	**22**
2.4.2	**Simple Mail Transfer Protocol (SMTP)**	**23**
2.4.3	**STARTTLS**	**25**
2.5	TRABALHOS RELACIONADOS	**27**
2.6	CONSIDERAÇÕES DO CAPÍTULO	**30**
3	**PROPOSTA**	**31**
3.1	CONSIDERAÇÕES DO CAPÍTULO	**32**
4	**CONCLUSÃO**	**34**
	REFERÊNCIAS	**35**

1 INTRODUÇÃO

O correio eletrônico, um dos serviços mais populares da Internet, tradicionalmente não oferece garantias de segurança. O transporte de email é realizado pelo protocolo SMTP (Simple Mail Transfer Protocol) (KLENSIN, 2008), que não possui mecanismos criptográficos nativos. O STARTTLS (HOFFMAN, 2002) (DUKHOVNI, 2014) é uma extensão para o SMTP que permite que servidores estabeleçam uma sessão TLS (Transport Layer Security) (RESCORLA, 2018) para garantir confidencialidade, integridade e autenticação do tráfego SMTP. O STARTTLS oferece um mecanismo criptográfico dito oportunista (DUKHOVNI, 2014): no início de uma sessão SMTP não criptografada, os servidores podem negociar o estabelecimento da sessão TLS, desde que ambos suportem essa opção. Existem dois problemas principais associados ao STARTTLS (BAATEN, 2019):

- Ataque de *downgrade*: como os pares trocam informações dentro de uma sessão SMTP desprotegida para saber se podem ou não iniciar uma sessão TLS, um atacante capaz de realizar um ataque de homem no meio (*man-in-the-middle*, MITM) pode simplesmente eliminar as mensagens que sinalizam o suporte a STARTTLS para forçar o uso de SMTP sem criptografia, sem que seja possível detectar essa supressão.

- Redirecionamento de tráfego: como normalmente a autenticidade dos certificados TLS dos servidores de email não é validada, um atacante capaz de realizar um MITM pode fornecer um certificado forjado e com isso interpor-se entre os servidores, violando as propriedades de segurança do TLS.

Uma solução para esses problemas é a adoção do padrão DANE para SMTP (DUKHOVNI; HARDAKER, 2015b), que usa registros TLSA no DNS (HOFFMAN; SCHLYTER, 2012) para sinalizar de forma segura que um servidor de email suporta STARTTLS e para publicar informações que permitem que servidores SMTP validem os certificados TLS recebidos durante o estabelecimento de uma sessão TLS. Com isso, as ameaças mencionadas acima são mitigadas. O DANE (DNS-based Authentication of Named Entities) (HOFFMAN; SCHLYTER, 2012) (DUKHOVNI; HARDAKER, 2015a), permite que o responsável por um domínio DNS publique um registro TLSA com informações sobre o certificado TLS usado por um dado servidor, como um servidor HTTPS ou SMTP+STARTTLS. A integridade e a autenticidade do registro TLSA são garantidas pelas extensões de segurança do DNS (DNSSEC) (ARENDS et al., 2005a). Os certificados publicados via DANE são emitidos pelo próprio servidor TLS, sem depender de uma autoridade certificadora (AC); isso mitiga uma ameaça clássica da infraestrutura de chaves públicas (PKI) do TLS, que é a possibilidade de que uma AC emita certificados válidos para qualquer nome (CLARK; OORSCHOT, 2013). Assim, para validar um certificado TLS apresentado por um servidor, um cliente pode (1) obter o registro TLSA do servidor, (2) validar esse registro usando as assinaturas do DNSSEC, e (3) verificar se o certificado é consistente com o registro TLSA. O DANE só pode funcionar corretamente quando todos os principais cumprem

suas responsabilidades, são eles: servidores TLS apresentando certificados, servidores DNS que publicam registros TLSA, clientes DNS validando respostas DNS usando DNSSEC e clientes TLS verificando certificados usando registros TLSA. Infelizmente, a complexidade do DANE leva a muitas oportunidades de erros de configuração. Por exemplo, no lado do servidor, podem conter erros DNSSEC nos registros TLSA, como assinaturas expiradas ou os certificados serem inconsistentes com os registros TLSA publicados. Já no lado do cliente, os resolvedores de DNS podem não validar os registros TLSA adequadamente ou aplicativos TLS com erros, de modo que não se preocupam em verificar a validade dos certificados (LEE et al., 2020).

1.1 JUSTIFICATIVA

Para entender o estado atual do ecossistema DANE para email na Internet, Lee (2020) realizou um estudo de medições envolvendo os domínios de segundo nível sob os domínios genéricos `.com`, `.net` e `.org`, e sob os domínios nacionais `.nl` e `.se` (respectivamente Holanda e Suécia). Ao longo de 24 meses, foram coletados dados sobre registros MX (que indicam servidores de email) e TLSA publicados nesses domínios, e também os certificados TLS apresentados pelos servidores SMTP encontrados. O estudo descobriu que 35% dos registros TLSA não podiam ser validados por conta de registros DNSSEC ausentes ou incorretos, e que 3.7% dos certificados eram inconsistentes com os registros TLSA correspondentes. O estudo também analisou aspectos do SMTP+DANE do ponto de vista do cliente, constatando que, apesar dos padrões serem suportados por software DNS e SMTP de código aberto, eles ainda são pouco adotados por provedores populares de email.

Não existem atualmente dados disponíveis sobre a adoção de DANE para email na Internet brasileira. O domínio `.br` não aparece nas estatísticas sobre SMTP+DANE (KNUBBEN, 2021). Este trabalho de conclusão de curso visa a preencher esta lacuna, replicando o estudo de (LEE et al., 2020) para o domínio `.br`. O objetivo primário é entender a adoção do DANE para proteção de email na Internet. Um objetivo secundário é testar a reprodutibilidade dos artefatos disponibilizados por (LEE et al., 2020). A reprodutibilidade tem sido um aspecto destacado e valorizado para promover pesquisa científica de qualidade (VITEK; KALIBERA, 2011) (PENG, 2011).

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Este trabalho tem como objetivo realizar um estudo de medições sobre o ecossistema do DANE para email na Internet brasileira.

1.2.2 Objetivos Específicos

- Realizar uma revisão bibliográfica abrangendo DNS, DANE, email e trabalhos relacionados;
- Replicar a infraestrutura de medições disponibilizada por (LEE et al., 2020) em um ambiente local;
- Coletar dados sobre o uso do DANE na Internet brasileira;
- Analisar os dados coletados.

1.3 METODOLOGIA

Visando a realização dos objetivos propostos na Seção 2, as atividades foram divididas da seguinte forma:

1. Revisão Bibliográfica;
2. Replicar a infraestrutura de medições em ambiente local;
3. Coletar os dados;
4. Análise dos dados;
5. Escrita da monografia.

1.4 ORGANIZAÇÃO DO TEXTO

Este documento está organizado da seguinte forma: o Capítulo 2 apresenta uma revisão da literatura, contemplando conceitos sobre DNS, DNSSEC, DANE, SMTP, STARTTLS e os trabalhos relacionados. No Capítulo 3 é apresentado a proposta desse trabalho de conclusão de curso seguindo pela conclusão no Capítulo 4.

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo tem como objetivo apresentar os conceitos necessários para o entendimento deste trabalho. A Seção 2.1 discorre sobre o conceito de DNS. A Seção 2.2 trata sobre o uso das Extensões de Segurança do Sistema de Nomes de Domínio (DNSSEC). A Seção 2.3 introduz o protocolo DANE. A Seção 2.4 trata do protocolo SMTP e STARTTLS. A Seção 2.5 apresenta alguns trabalhos relacionados. Por fim, a Secrita de forma impessoal eção 2.6 trás as considerações finais sobre o capítulo.

2.1 DOMAIN NAME SYSTEM (DNS)

Um conceito de suma importância se tratando da Internet é o endereço IP, que é um número de 32 bits (128 bits no IPv6) e que serve como por exemplo para identificar um *host*. Os endereços IP são convenientes para os computadores mas não para os os usuários, tanto pela dificuldade de memorização quanto pela possibilidade de que *hosts* mudem de endereço (devido a uma reconfiguração de topologia ou mudança de provedor, por exemplo). Sendo assim, surgiu a necessidade de nomear esses números complexos de 32 bits.

Na década de 70, quando a Internet ainda era conhecida como ARPANET e possuía apenas algumas centenas de *hosts*, tinha-se um arquivo `HOSTS.TXT` que continha o mapeamento de nome para endereço de todo *host* na ARPANET (LIU; ALBITZ, 2006). Esse arquivo era mantido pelo SRI-NIC (*Stanford Research Institute Network Information Center*) e conforme o crescimento da ARPANET, se tornou inviável o uso do mesmo, assim necessitando uma solução melhor para resolver todas as associações nome-endereço. Como solução, em 1984 foi adotado o Domain Name System (DNS), descrito originalmente nas RFC 882 (MOCKAPETRIS, 1983a) e 883 (MOCKAPETRIS, 1983b)

2.1.1 O que é DNS

De acordo com (COSTA, 2006) o DNS é um protocolo da camada de aplicação com intuito de auxiliar os outros protocolos de aplicação, fazendo a tradução de endereços numéricos e nomes, por exemplo, permitir que informações textuais como o nome de um site (*www.example.com.br*) sejam mais familiares ao invés de um endereço IP (192.0.2.1).

O DNS é um banco de dados distribuído (LIU; ALBITZ, 2006). Essa estrutura possibilita que ocorra o controle local de cada segmento do repositório, mas de maneira que a informação em cada segmento encontre-se disponível para toda a Internet num esquema de cliente/servidor. Dessa forma basta que administradores gerenciem seus segmentos e servidores de nomes para manter as informações atualizadas para o restante da rede. Isso permite também a redistribuição do gerenciamento das informações de domínios em várias organizações especializadas (SCHUBA, 1993). Robustez e desempenho são alcançados através de replicação dos dados e armazenamento local em *cache*.

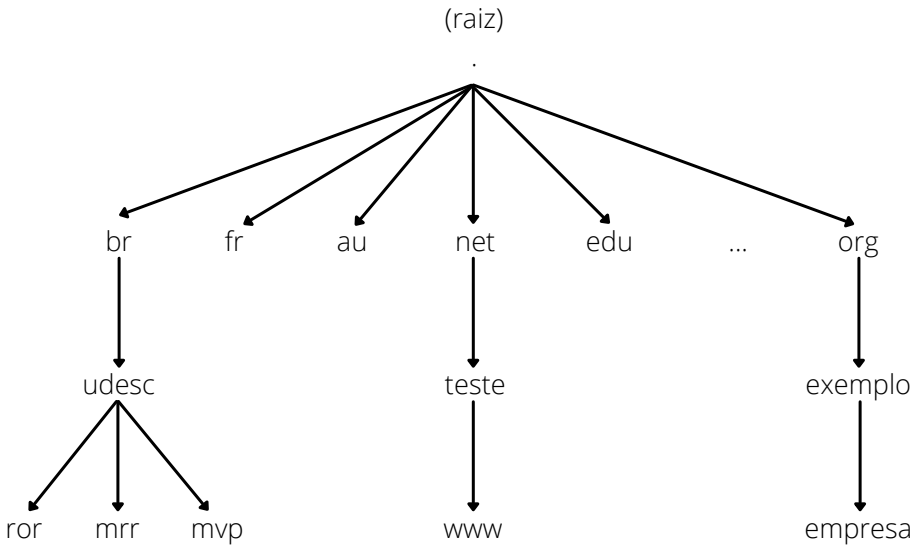
Os programas chamados de servidores de nomes formam parte do mecanismo cliente/servidor do DNS. Os servidores de nomes contêm informações sobre alguns segmentos do banco de dados e as disponibilizam aos clientes, chamados de *resolvers* ou resolvedores. Os resolvedores geralmente são apenas rotinas de biblioteca que criam requisições e consultas e as enviam através de uma rede para um servidor de nomes (LIU; ALBITZ, 2006).

Foi visto nessa subseção como o DNS funciona e a sua importância, pois é ele que facilita na hora de escrevermos na URL o nome de um site ao invés de digitarmos um número complicado. Nas próximas subseções, será discutido sobre os componentes do DNS, veremos que ele é formado por: espaço de nomes, registros de recurso, mensagens DNS, clientes (resolvedores) e servidores DNS.

2.1.2 Espaço de nomes do DNS

O espaço de nomes do DNS é hierárquico, com estrutura em formato de árvore invertida, onde a raiz da árvore é o domínio raiz, e seus filhos são os domínios de primeiro nível, que podem ou não conter diversos níveis de subdomínios (MOCKAPETRIS, 1987b).Um nome de domínio completo é representado pela concatenação dos rótulos de cada nó da árvore, desde a folha (que representa a entidade nomeada) até a raiz. Um ponto é usado como separador entre os rótulos, sendo que o domínio raiz é representado por um rótulo vazio. Um domínio é uma subárvore desse espaço de nomes (LIU; ALBITZ, 2006). Observando a Figura 1, utilizamos o exemplo `mrr.udesc.br.` que é um nome de domínio, onde o ponto ao final do nome separa o domínio `br` do domínio raiz porém muitas vezes o ponto final é omitido.

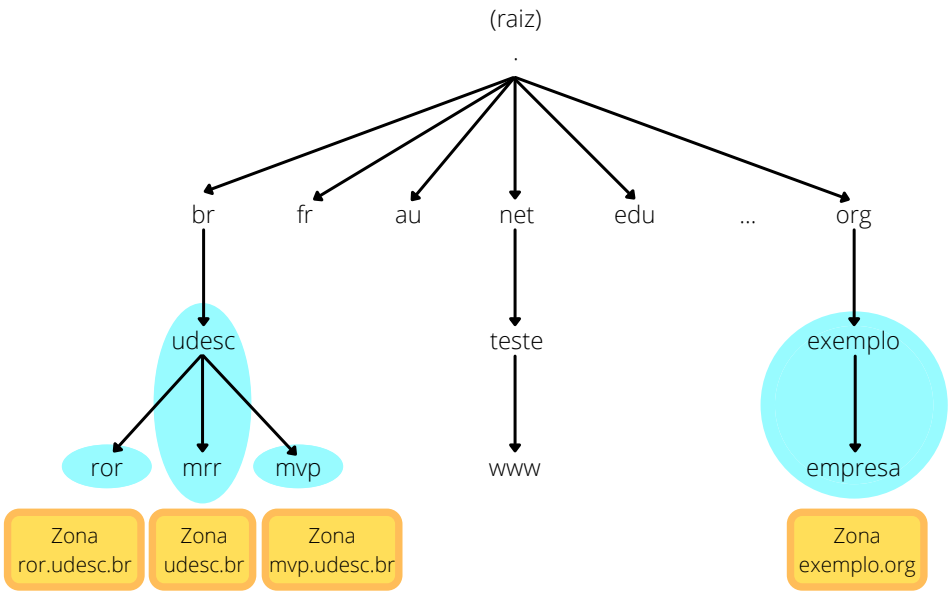
Figura 1 – Espaço de nomes do DNS



Fonte: Imagem elaborada pelo autor

O conceito de zona é muito parecido com o de domínio: um domínio é uma subárvore do espaço de nomes, enquanto uma zona é a parte do espaço de nomes gerenciada por um servidor de nomes (MOCKAPETRIS, 1987b). Quando um servidor de nomes controla todos os nomes de um domínio, ou seja, quando não ocorrer delegação de autoridade, a zona e o domínio são idênticos. Quando algumas partes de um domínio são delegadas para outros servidores, a zona contém apenas as partes não delegadas desse domínio. Usando a mesma árvore de domínios da Figura 1, podemos criar uma situação hipotética de que o domínio `mvp.udesc.br` é administrado por um servidor de nomes diferente do que controla o domínio `udesc.br`, de modo que ele não faz parte da zona `udesc.br`. Por sua vez, `mrr.udesc.br` não é administrado de forma independente do domínio `udesc.br` e, portanto, faz parte da zona `udesc.br`, conforme ilustra a Figura 2.

Figura 2 – Delegação de Zonas



Fonte: Imagem elaborada pelo autor

Cada elemento da árvore DNS possui um item de dados associado, chamado de registro de recurso (*resource record*, RR) (MOCKAPETRIS, 1987b). Um RR é identificado por uma tripla < nome, tipo, classe >. O nome determina a sua localização na árvore DNS. Os tipos representam a variedade de informações que podem ser armazenadas para um dado domínio, os principais tipos estão listados na Tabela 1. A classe possibilita que RRs de mesmo tipo tenham formatos distintos caso se refiram a famílias de protocolos diferentes. A classe mais usada é a classe IN, que representa a Internet. Possui também outras classes como CH (antiga rede Chaosnet) e HS (*software* Hesiod). (LIU; ALBITZ, 2006) (MOCKAPETRIS, 1987b)

Um RR padrão é composto pelos seguintes campos abaixo e também os que já foram citados anteriormente como NAME, TYPE e CLASS que correspondem respectivamente o nome, o tipo e a classe (MOCKAPETRIS, 1987b):

Tabela 1 – Principais tipos de RRs

Tipo	**Definição**
A	endereço IPv4 para um domínio
AAAA	endereço IPv6 para um domínio
NS	servidor de nomes autoritativo para o domínio
TXT	texto de strings
SOA	início de autoridade
PTR	ponteiro para um outro nome de um domínio no espaço de domínios
CNAME	nome canônico para um apelido/sinônimo (alias) de um domínio/servidor
MX	servidor de emails para o domínio

Fonte: Tabela elaborada pelo autor

- TTL (*Time to Live*): É o tempo de vida do RR em segundos. Com um campo de 32 bits, o TTL é usado principalmente para determinar quanto tempo um RR pode permanecer em cache nos servidores de resolução de nomes. Esse valor é determinado pelo administrador da zona na qual se originou esse RR.
- RDLENGTH: Um campo de 16 bits que determina o tamanho em bytes de RDATA.
- RDATA: Podendo variar de acordo com o TYPE e a CLASS, o RDATA é uma sequência de bytes que descreve o recurso.

Tabela 2 – Formato de um RR

NAME
TYPE
CLASS
TTL
RDLENGTH
RDATA

Fonte: Tabela elaborada pelo autor

A Tabela 3 mostra alguns exemplos de RRs para uma zona hipotética `example.org`. Com esse exemplo é possível identificar que RRs de tipos diferentes possuem também formatos diferentes de informação como registros A carregam endereços IP, registros NS carregam nomes e registros MX carregam um nome e uma preferência.

Um conjunto de RRs com o mesmo nome, tipo e classe porém com dados diferentes (RDATA), é chamado de RRset (ELZ; BUSH, 1997). Na Tabela 6 segue um exemplo de RRset para <example.org, IN, NS>.

Tabela 3 – Exemplos de RRs

example.org.	IN	NS	ns.example.org.
alpha.example.org.	IN	A	192.0.2.1
www.example.org.	IN	CNAME	beta.example.org.
example.org.	IN	MX 10	alpha.example.org.
1.2.0.192.in-addr.arpa.	IN	PTR	alpha.example.org.

Fonte: Tabela elaborada pelo autor

Tabela 4 – Exemplos de RRset

example.org.	IN	NS	ns.example.org.
example.org.	IN	NS	ns.example.net.

Fonte: Tabela elaborada pelo autor

2.1.3 Resolução de nomes

Resolução de nomes é o processo de recuperação de dados armazenados no DNS, ou seja, um *host* obtém determinadas informações DNS sobre um dado nome. Para isso, aplicações recorrem a um cliente chamado de resolvedor (*resolver*) e um ou mais servidores DNS. Segundo (LIU; ALBITZ, 2006), resolvedor são os clientes que acessam servidores de nomes. O resolvedor se responsabiliza por:

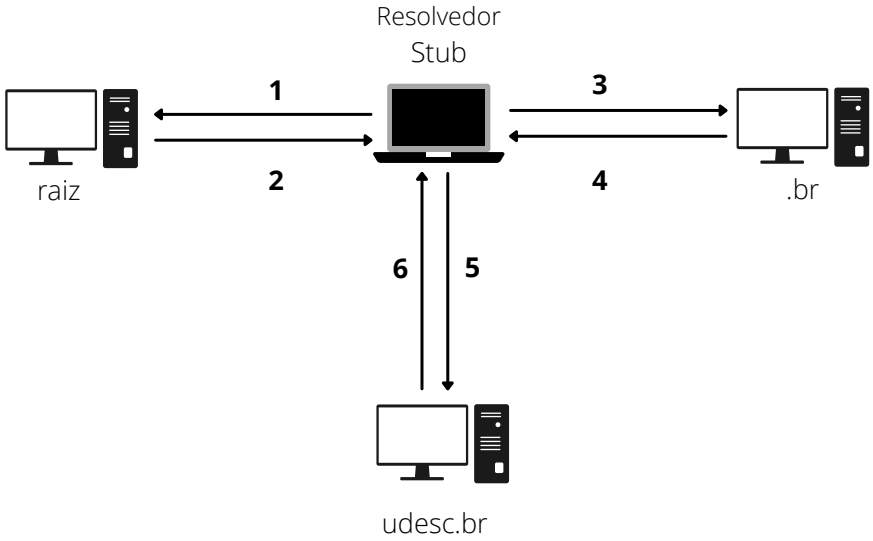
- Enviar consultas para os servidores DNS;
- Interpretar respostas (que podem ser registros de recursos ou um erro); e
- Devolver as informações aos programas que as solicitaram.

Como visto na Figura 2, existe a delegação de zonas, portanto diversas vezes é necessário visitar diversos servidores para resolver um determinado nome. Usando esse mesmo exemplo da Figura 2, o servidor de `udesc.br` não é responsável pelo nome `mvp.udesc.br`, isso porque o subdomínio `mvp` está em uma zona separada. Ao receber uma consulta por `mvp.udesc.br`, o servidor de `udesc.br` retorna os dados do servidor responsável pela zona.

A resolução de nomes pode ser feita de dois modos, iterativo e recursivo. No modo iterativo, o servidor DNS retorna ao cliente a melhor resposta de que dispõe, que pode ser a resposta final ou informações sobre os servidores de nomes mais próximos do nome desejado, de acordo com a hierarquia de zonas (MOCKAPETRIS, 1987b). Caso o servidor não retorne uma resposta final, o cliente dá sequência à resolução usando um ou mais dos servidores de nomes indicados. A Figura 3 exemplifica o processo de uma consulta DNS iterativa. Nesse exemplo, o resolvedor stub é que realiza as consultas para descobrir o IP do domínio `www.udesc.br`. Os passos realizados estão numerados, (1) o resolvedor stub envia a consulta DNS para encontrar o domínio `www.udesc.br` ao servidor autoritativo raiz . Porém o servidor raiz não conhece esse domínio, mas devolve uma resposta parcial (2) indicando o endereço do servidor `.br`. (3) O

Resolvedor stub consulta o servidor autoritativo `.br` e recebe (4) como resposta o endereço do servidor `udesc.br`. (5) É realizado a consulta então ao servidor autoritativo `udesc.br`, e como o `www.udesc.br` está na zona do servidor `udesc.br`, ele devolve (6) o endereço IP correto.

Figura 3 – Exemplo de consulta DNS iterativa



Fonte: Imagem elaborada pelo autor

Em uma resolução recursiva, o cliente envia a consulta para um servidor DNS local (dito resolvedor recursivo), que irá efetuar o processo iterativo de resolução (ou repassar a consulta para um outro recursivo) e apenas devolver a resposta final para o resolvedor stub.

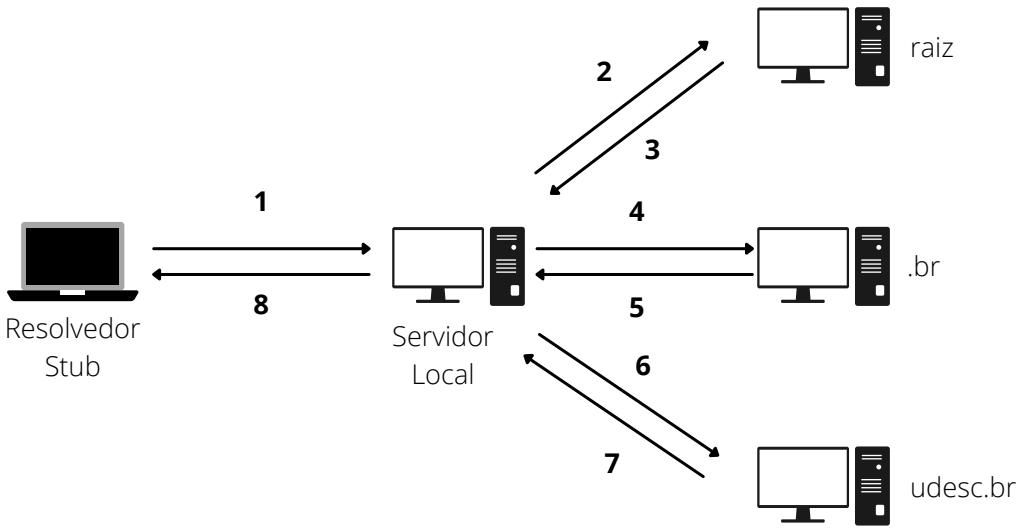
Normalmente, cada rede possui pelo menos um servidor capaz de atender a consultas recursivas. Utilizando o mesmo exemplo no modo iterativo, observamos na Figura 4 o resolvedor stub (1) se comunicando com o seu servidor de nomes local, que é o responsável por se comunicar com os outros servidores a fim de obter a resposta válida (ou um erro), o mesmo que ocorre no modo iterativo. Um problema decorrente de configuração permitir que qualquer máquina na Internet envie consultas para o servidor DNS recursivo de uma determinada rede. Os servidores que possuem esse problema são chamados de servidores DNS recursivos abertos, isso porque apenas o servidor autoritativo deve responder consultas de máquinas externas (CERT.BR, 2013).

2.1.4 Formato de mensagens DNS

Ao realizar uma conversa entre cliente e servidor, realizando a troca de informações, usa-se mensagens de requisição e de resposta, e tanto as mensagens de requisição quanto de resposta possuem um mesmo formato, possuindo até cinco seções individuais.

As duas primeiras seções, a seção Cabeçalho (*Header*) e a seção de Pergunta (*Question*), são encontradas tanto na mensagem de requisição quanto na de resposta. As seções seguintes

Figura 4 – Exemplo de consulta DNS recursiva



Fonte: Imagem elaborada pelo autor

são usadas apenas em mensagens de resposta DNS, que são as seções de Resposta (*Answer*), Autoridade (*Authority*) e Adicional (*Additional*) (MOCKAPETRIS, 1987a).

O Cabeçalho possui 12 bytes e inclui campos que especificam quais das seções restantes estão presentes na mensagem, e também indica se é uma pergunta ou uma resposta. A seção Pergunta contém campos que descrevem um pergunta a um servidor de nomes. Esses campos são: QTYPE (tipo), QCLASS (classe) e QNAME (nome de domínio). As últimas três seções (Resposta, Autoridade e Adicional) possuem o mesmo formato, uma lista de registros de recursos (RRs) concatenados, podendo ser vazia. A seção de Resposta contém RRs que respondem a pergunta. A seção Autoridade contém RRs que apontam para servidores de nomes com autoridade sobre o nome de domínio de interesse. A seção Adicional contém RRs que se relacionam com a pergunta, mas não são respostas diretas para a pergunta. Para um melhor entendimento, segue a Tabela

Tabela 5 – Exemplo de uma mensagem de consulta DNS

`Cabeçalho`	`ID=123`
`Pergunta`	`QNAME=alpha.example.org., QTYPE=A QCLASS=IN`
`Resposta`	
`Autoridade`	
`Adicional`	

Fonte: Tabela elaborada pelo autor

As mensagens DNS entre um cliente e um servidor são encaminhadas como mensagens de aplicação utilizando o protocolo UDP e a porta 53 (KUROSE, 2010).

Tabela 6 – Exemplo de uma mensagem de resposta DNS

Cabeçalho	ID=123
Pergunta	
Resposta	alpha.example.org. IN A 192.0.2.1
Autoridade	example.org. IN NS ns.example.org. example.org. IN NS ns.example.net.
Adicional	ns.example.org. IN A 192.0.2.3 ns.example.net. IN A 10.4.8.25

Fonte: Tabela elaborada pelo autor

2.2 DNSSEC

As Extensões de Segurança do Sistema de Nomes de Domínio (DNSSEC) adicionam origem de dados autenticação e integridade de dados para o Sistema de Nome de Domínio (DNS) (ARENDS et al., 2005a). Segundo (THO, 2017), DNSSEC é um desenvolvimento de segurança DNS executado em cima da infraestrutura DNS comum que fornece propriedades de autenticação e integridade para respostas DNS. Isso ajuda o DNS a resistir a ataques que dependem do envio de dados falsificados, como envenenamento de cache. No entanto, é importante notar que DNSSEC não fornece confidencialidade, uma vez que todas as mensagens de DNS ainda são enviadas em texto não criptografado.

O comportamento usual do DNS de enviar uma consulta inteira ou resposta em um único pacote UDP não assinado e não criptografado torna os ataques particularmente fáceis para qualquer atacante com a capacidade de interceptar pacotes em uma rede compartilhada ou de trânsito (ATKINS; AUSTEIN, 2004), desta forma, o DNSSEC vem com uma proposta de resolver esses problemas de vulnerabilidade do DNS, que por sua vez, possui um tráfego suscetível a manipulação.

A solução, utilizada pelo DNSSEC, está em fazer a assinatura digital das informações da zona utilizando chaves assimétricas (pública e privada), garantindo integridade e autenticidade das informações e provendo então segurança na resolução de endereços (REGISTRO.BR, 2021). Essas assinaturas digitais são armazenadas nos *nameservers* do DNS juntamente com tipos comuns de registros, como o A, o AAAA, o MX, o CNAME etc. Ao verificar a assinatura dos registros em uma resposta e a cadeia de confiança que garante a chave de assinatura, é possível acreditar que tais registros são legítimos e não forma manipulados indevidamente (CLOUDFLARE, 2021).

Para permitir a validação de assinaturas, o DNSSEC adiciona alguns novos tipos de registros DNS (ARENDS et al., 2005c):

- RRSIG: Contém a assinatura digital de um RRset assinado com a chave privada que compartilha nome / tipo / classe;
- DNSKEY: contém uma chave pública assinada, usada para assinar um conjunto de

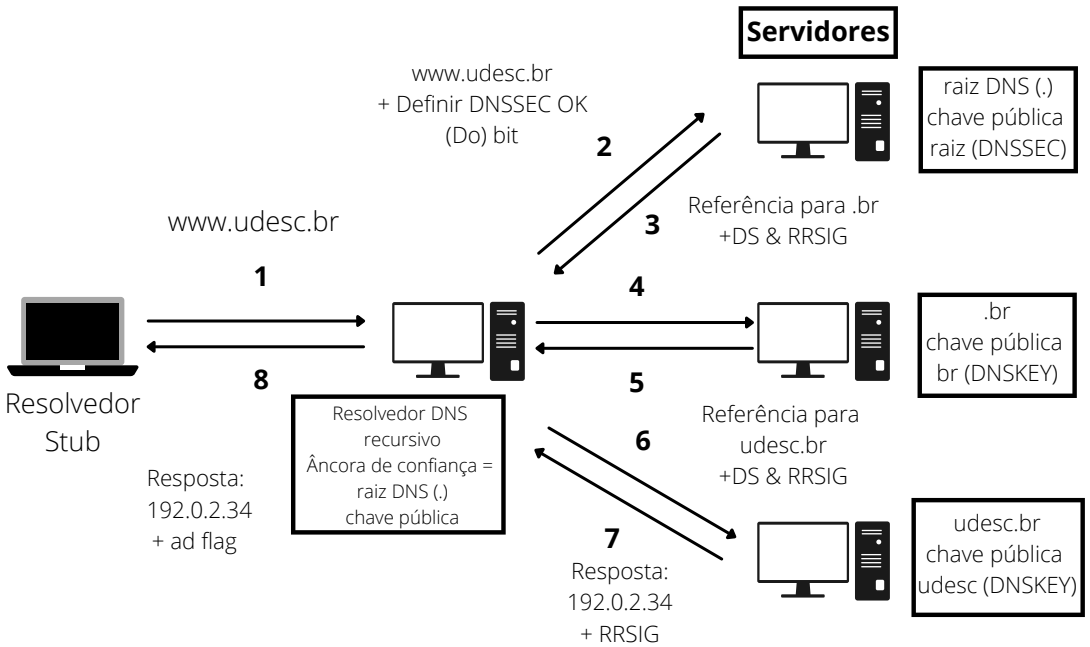
registros de recursos (RRset);

- NSEC, NSEC3: registros NSEC e NSEC3 fornecem uma prova autenticada de inexistência, ou seja, eles permitem certificar que o RR consultado não existe. Como registros NSEC permitem trivialmente enumerar todos os nomes existentes em uma zona (ARENDS et al., 2005a), a RFC 5155 (LAURIE et al., 2008) introduziu os registros NSEC3 para mitiigar o problema. No entanto, os registros NSEC3 não impedem a enumeração de zonas, apenas dificultam a tarefa (GOLDBERG et al., 2015).

- DS: contém o hash de um registro DNSKEY;

Um conceito importante abordado pelo DNSSEC é a cadeia de confiança, que permite certificar a chave pública (registro DNSKEY) usada para verificar a assinatura de uma zona. O mecanismo que o DNSSEC funciona de maneira que uma zona pai pode hospedar um ou mais registros DS e cada registro DS se refere a uma chave pública em uma zona filha. Ou seja, se a zona pai for confiável, os validadores podem confiar que as chaves públicas na zona filha são confiáveis também. Um resolvedor precisa ter uma âncora de confiança, tipicamente a chave pública do domínio raiz, para validar recursivamente a cadeia de chaves e assinaturas. Essa chave pública pode ser distribuída junto com o software DNS ou obtida pela web (ABLEY et al., 2016).

Figura 5 – Exemplo de um cliente fazendo uma consulta DNS para `www.udesc.br` usando um resolvedor DNS recursivo com reconhecimento de segurança



Fonte: Imagem adaptada de (SANDELIN, 2017)

Na Figura 5 temos um exemplo de resolução recursivo do registro `www.udesc.br A`. O recursivo dispõe da chave pública da zona raiz como âncora de confiança, e seu cache está vazio. No passo (1), o resolvedor *stub* envia uma consulta para o recursivo, que é o responsável pela

validação das assinaturas do DNSSEC. Os passos (2) e (3) mostram a interação do recursivo com um dos servidores DNS da raiz. O recursivo recupera o RRset DNSKEY da raiz com o respectivo RRSIG, e verifica se a assinatura está correta, dentro do prazo de validade, e em conformidade com a âncora de confiança. Depois, ele consulta os registros NS e DS do domínio `.br`, e verifica se a assinatura do registro DS está correta (os registros NS não são assinados, assim como os registros A eventualmente retornados na seção adicional da resposta). Nos passos (4) e (5) ocorre a interação do recursivo com o servidor do `.br`. Primeiro é recuperado e validado o RRset DNSKEY do `.br`, que deve ter uma assinatura válida e estar de acordo com o registro DS na raiz. Depois são recuperados os registros NS e DS do domínio `udesc.br`, e a assinatura do DS é validada. Os passos (6) e (7) mostram a interação do recursivo com o servidor de `udesc.br`. Inicialmente o RRset DNSKEY é recuperado e validado, e na sequência o recursivo obtém o registro A para `www.udesc.br` com o respectivo RRSIG, e valida a assinatura. Finalmente, no passo (8) o recursivo retorna a resposta da consulta para o resolvedor *stub*; a flag AD no cabeçalho da resposta indica que a resposta foi autenticada pelo recursivo. Os registros validados pelo recursivo ao longo do processo (por exemplo, os registros DNSKEY) podem ser armazenados em cache e reaproveitados em consultas subsequentes. Por exemplo, se o recursivo na sequência recebesse uma consulta por `udesc.br MX`, ele poderia entrar em contato diretamente com o servidor de nomes de `udesc.br` e pedir apenas o registro MX (e seu respectivo RRSIG), validando a assinatura com a chave no cache.

Um resolvedor DNSSEC pode determinar o resultado de uma resolução como seguro, inseguro, falso ou indeterminado, conforme explicado nas RFCs 4033 (ARENDS et al., 2005a) e 4035 (ARENDS et al., 2005b):

- Seguro: quando o resolvedor pode seguir a cadeia de confiança e verificar com sucesso um RRset com seu RRSIG correspondente.
- Inseguro: quando um resolvedor não pode seguir a cadeia de confiança de uma âncora de confiança para um RRset devido à inexistência de um registro DS.
- Falso: quando um resolvedor deveria ser capaz de seguir a cadeia de confiança de uma âncora de confiança para um RRset, entretanto, a validação falha por alguns motivos, como assinaturas ausentes ou assinaturas expiradas. Isso pode acontecer devido a erros de configuração, corrupção de dados ou ataques reais.
- Indeterminado: um resolvedor não pode determinar que uma parte específica da hierarquia DNS é segura devido à falta de uma âncora de confiança.

No exemplo da Figura 5, se todas as chaves e assinaturas fossem válidas, o resultado seria uma resolução segura.

2.3 DANE

O protocolo Transport Layer Security (TLS) (RESCORLA, 2018) oferece garantia de confidencialidade, integridade e autenticação de origem para diversas aplicações que usam canais TCP, como HTTP e SMTP. O TLS usa criptografia de chave pública, e depende de uma infraestrutura de chaves públicas (*Public-Key Infrastructure*, PKI) e certificados para vincular entidades (clientes e servidores) às suas chaves públicas. Os certificados são normalmente emitidos por autoridades de certificação (ACs), de forma hierárquica. Os clientes são alimentados com uma lista de chaves públicas que são consideradas confiáveis (as chamadas âncoras de confiança) e que são usadas para validar uma cadeia de certificados. A premissa da PKI do TLS é que qualquer AC pode emitir certificados para qualquer entidade: um certificado será considerado legítimo se puder ser validado usando uma das âncoras de confiança (CLARK; OORSCHOT, 2013) (DÍAZ-SÁNCHEZ, 2019). Se uma AC for enganada a assinar um certificado inválido, ou tiver sua chave privada comprometida, a premissa será violada. Para resolver esse problema, o protocolo *DNS-based Authentication of Named Entities* (DANE) foi proposto para oferecer suporte à validação de certificados TLS sem depender de terceiros confiáveis, como ACs (HOFFMAN; SCHLYTER, 2012).

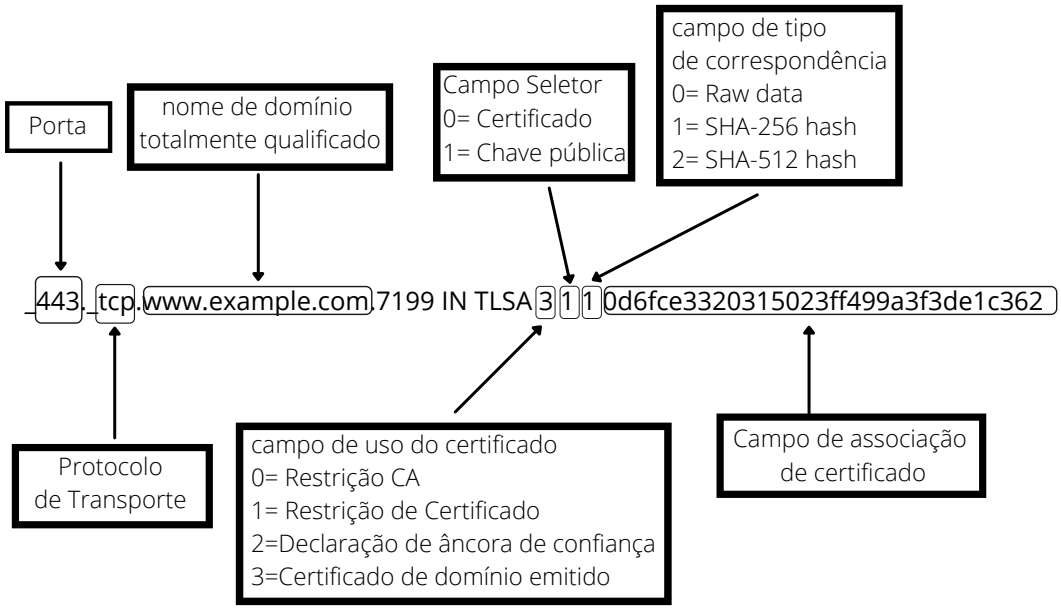
A autenticação baseada em DNS de entidades nomeadas (DANE) depende do DNSSEC para autenticar os registros DNS DANE com o propósito de associar nomes de domínio com credenciais (certificados PKI) (DÍAZ-SÁNCHEZ, 2019). DANE é proposto na RFC 6698 (HOFFMAN; SCHLYTER, 2012) como uma forma de autenticar entidades de cliente e servidor TLS sem uma autoridade de certificação (AC). Ele é atualizado com orientações operacionais e de implantação no RFC 7671 (DUKHOVNI; HARDAKER, 2015a) . O uso específico do DANE para SMTP é definido na RFC 7672 (DUKHOVNI; HARDAKER, 2015b).

O DANE permite que os proprietários de domínio incluam informações sobre as credenciais de autenticação de seus serviços permanentes em seu DNS. Considerando que o DNS é normalmente consultado para resolução de nomes pelo cliente antes de se conectar ao servidor, o cliente pode receber informações relativas à credencial que deve ser recebida do servidor durante o *handshake* TLS antes de realmente se conectar ao servidor. Isso evita que certificados maliciosos sejam usados para personificar servidores gerenciados pelo proprietário do domínio (DÍAZ-SÁNCHEZ, 2019) (CLARK; OORSCHOT, 2013).

O DANE define um novo registro DNS chamado TLSA, que permite ao administrador de uma zona DNS associar um certificado ou uma chave pública a um servidor ou âncora de confiança para um determinado serviço de rede (HOFFMAN; SCHLYTER, 2012). O nome usado para o RR indica a porta e o protocolo de transporte em que o registro é válido; por exemplo, `_443._tcp.www.example.com` indica o registro TLSA para o serviço na porta TCP 443 do servidor `www.example.com`. Caso o mesmo *host* tenha mais serviços com TLS em outras portas, ele irá precisar definir RRs TLSA adicionais. Um registro TLSA possui quatro campos: uso do certificado, seletor, tipo de correspondência e associação do certificado (HOFFMAN;

SCHLYTER, 2012). Na Figura 6 temos um exemplo de um Registro de Recurso TLSA para melhor entendimento de cada campo RDATA.

Figura 6 – Exemplo de um Registro de Recurso DANE TLSA



Fonte: Adaptado de (SANDELIN, 2017)

O campo de uso do certificado pode ter o valor 0, 1, 2 ou 3. Quando o valor 0 (PKIX-TA) é utilizado, isso especifica um certificado AC ou a chave pública de tal certificado que precisa estar presente em qualquer um dos caminhos de certificação PKIX para o certificado fornecido pelo servidor. Esse tipo de uso também é conhecido como restrição de AC, pois restringe quais ACs podem ser usadas para emitir certificados para um determinado serviço (HOFFMAN; SCHLYTER, 2012). Quando o valor 1 (PKIX-EE) é utilizado, o RR especifica um certificado de entidade final ou chave pública de tal certificado que precisa corresponder ao certificado fornecido pelo servidor. Esse tipo de uso também é conhecido como restrição de certificado de serviço, pois restringe qual certificado de entidade final pode ser usado por um determinado serviço (HOFFMAN; SCHLYTER, 2012). O valor 2 (DANE-TA) especifica um certificado ou chave pública que deve ser usado como âncora de confiança. Esse tipo pode ser usado quando um domínio emite um certificado sob sua própria AC e é improvável que essa AC esteja presente na lista de âncoras de confiança dos clientes. Este tipo também é conhecido como uma declaração de âncora de confiança (HOFFMAN; SCHLYTER, 2012). O valor 3 (DANE-EE) especifica um certificado ou chave pública que deve corresponder ao certificado fornecido pelo servidor. A diferença com o uso do certificado 3 dos demais, é que no caso dos demais valores o certificado deve, além de corresponder ao registro TLSA, também passar na validação da infraestrutura de chave pública para certificados X.509 (PKIX) que não é realizada no caso de uso 3. Este tipo de uso também é conhecido como certificado emitido por domínio, pois permite que um administrador de DNS emita certificados para um serviço sem envolver uma CA de terceiros

(HOFFMAN; SCHLYTER, 2012). Quando o uso for PKIX-TA(0) ou PKIX-EE(1), a validação do certificado depende de âncoras de confiança adicionais pré-configuradas que tenham confiança tanto de clientes quanto de servidores TLS (DUKHOVNI; HARDAKER, 2015a). Quando o uso for DANE-TA(2) ou DANE-EE(3), âncoras de confiança adicionais não são necessárias, e os registros TLSA são suficientes para validar o certificado TLS do servidor. A recomendação é que um serviço use apenas DANE-TA(2) e DANE-EE(3), evitando que os clientes precisem manter uma lista atualizada de âncoras de confiança para validar os certificados recebidos (DUKHOVNI; HARDAKER, 2015a).

O campo do seletor especifica qual parte do certificado do servidor corresponderá aos dados de associação no registro TLSA. O campo seletor pode ter o valor 0 ou 1, onde um valor 0 indica que o certificado completo deve corresponder e o valor 1 indica que apenas a chave pública deve corresponder (HOFFMAN; SCHLYTER, 2012).

O campo de tipo de correspondência pode ter os valores 0, 1 ou 2, onde 0 é a correspondência exata, i.e. todo o certificado. O tipo de correspondência 1 indica que um hash SHA-256 precisa ser aplicado e o tipo 2 é um hash SHA-512 (HOFFMAN; SCHLYTER, 2012).

O campo de dados de associação de certificado contém os dados que devem corresponder conforme ditado pelo seletor e o campo de tipo de correspondência, ou seja, o certificado completo, um hash SHA-256 ou um hash SHA-512 (HOFFMAN; SCHLYTER, 2012).

No exemplo da Figura 6, `0d6fce...` é o hash SHA-256 (tipo=2) da chave pública (seletor=1) contida no certificado (uso=3) fornecido pelo servidor que usa a porta TCP 443 no *host* <www.example.com>; como uso=3 (DANE-EE), esse certificado não deve ser validado usando a cadeia de certificação da PKI X.509, sua validade é garantida pela própria PKI do DNSSEC.

2.4 SMTP E STARTTLS

Nessa Seção será tratado sobre o Simple Mail Transfer Protocol (SMTP), tal protocolo é responsável pelo envio de mensagens de correio eletrônico através da internet entre um emissor e um receptor. Pelo fato desse protocolo ser um protocolo que utiliza o envio de email em texto simples, o uso do STARTTLS se faz necessário, que é uma extensão que adiciona autenticação e confidencialidade na comunicação entre agentes SMTP.

2.4.1 Modelo de processamento de email na Internet

O envio de correio eletrônico (email) pela Internet envolve diversos atores e protocolos (KUROSE, 2010) (CROCKER, 2009) (ROSE et al., 2019). Como ilustra a Figura 7, uma mensagem pode ser processada por vários servidores no seu trajeto entre os usuários de origem e destino. O usuário de origem escreve uma mensagem em um cliente de email, chamado de agente do usuário (MUA, *mail user agent*). O cliente pode tanto ser uma aplicação *standalone*, como Mozilla Thunderbird ou Microsoft Outlook, quanto uma aplicação de *webmail*. Essa

mensagem é enviada para um servidor de email chamado de agente de submissão (MSA, *mail submission agent*). O MSA é operado pelo provedor de email do usuário, e é responsável por autenticar o usuário, garantir que a mensagem respeite o formato definido nas especificações relevantes, e implementar outras políticas locais (como limitação do volume de mensagens enviadas) (CROCKER, 2009). Caso o MSA aceite a mensagem, ela é entregue a um agente de transferência (MTA, *mail transfer agent*), que é um servidor responsável por encaminhar a mensagem em direção ao destinatário. O papel de MTA pode ser desempenhado pelo mesmo servidor que hospeda o MSA, ou por um servidor separado. Arquiteturas usando múltiplos MSAs e MTAs para atender a requisitos de desempenho, segurança e disponibilidade também são possíveis. Uma mensagem pode percorrer diversos MTAs até chegar ao MTA de saída, que será responsável pelo envio para o provedor de email do destinatário.

Para descobrir o servidor de email do provedor de destino, o MTA de saída usa o DNS, pesquisando os registros MX (*Mail eXchange*) do domínio do receptor (KLENSIN, 2008). Esses registros indicam quais são os servidores de email responsáveis por um domínio, e qual sua ordem de preferência (isto é, em que sequência eles devem ser contactados). O MTA de origem seleciona o MTA de destino mais adequado, de acordo com os registros MX, e envia a mensagem para esse servidor. Ao aceitar a mensagem, o MTA de destino a repassa a um agente de entrega (*mail delivery agent*, MDA), que é responsável pelo gerenciamento das caixas postais dos usuários. A última etapa do processo consiste na interação do MUA e do MDA do destinatário para autenticar o usuário e transferir ou exibir as mensagens na sua caixa postal.

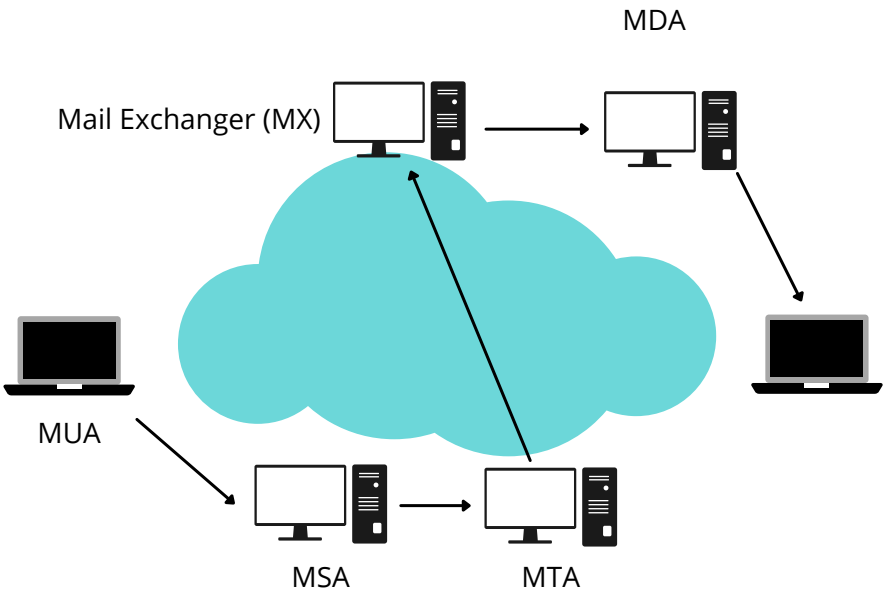
Vários protocolos são usados nessa arquitetura. As comunicações que envolvem um MTA (seja MSA-MTA, MTA-MTA ou MTA-MDA) usam o Simple Mail Transfer Protocol (SMTP) (KLENSIN, 2008). As comunicações MUA-MSA usam o Submission, que é um subconjunto do SMTP com algumas restrições de uso adicionais (GELLENS; KLENSIN, 2011). As comunicações MDA-MUA usam protocolos como POP (MYERS; ROSE, 1996) e IMAP (CRISPIN, 2003). Como o foco deste trabalho é a comunicação entre MTAs, apenas o protocolo SMTP será apresentado na próxima seção.

2.4.2 Simple Mail Transfer Protocol (SMTP)

Como visto as mensagens de email são transferidas de um servidor de email para outro (ou de um MUA para MSA / MTA) usando SMTP. O SMTP foi originalmente especificado em 1982 na RFC 821 (POSTEL, 1982) e passou por várias revisões, sendo a mais recente o RFC 5321 (KLENSIN, 2008). O SMTP é um protocolo cliente-servidor baseado em texto em que o cliente (remetente do email) contata o servidor (MTA do próximo salto) e emite um conjunto de comandos para informar ao servidor sobre a mensagem a ser enviada e, em seguida, transmite a própria mensagem. A maioria desses comandos são mensagens de texto ASCII enviadas pelo cliente e um código de retorno resultante (também texto ASCII) retornado pelo servidor (ROSE et al., 2019).

A Figura 8 apresenta um exemplo simples de transação SMTP, em que uma usuária Alice

Figura 7 – Funcionamento de um envio de email



Fonte: Imagem criada pelo autor

(`alice@gmail.com`) envia uma mensagem para Bob (`bob@hotmail.com`). As linhas iniciadas por `C:` indicam tráfego enviado pelo cliente, e as linhas iniciadas por `S:` indicam o tráfego enviado pelo servidor. Logo na linha 1 o servidor identificando com "220 mx.google.com Simple Mail Transfer Service Ready" que o cliente de email pode prosseguir com a comunicação. Logo em seguida (linha 2) o cliente envia o comando "EHLO hotmail.com" informando que gostaria de utilizar SMTP estendido; o servidor então confirma (linha 3) a requisição do cliente com "250-mx.google.com at your service", e informa uma lista de parâmetros suportados (linhas 4–8). O cliente indica os endereços do remetente (linha 9) e do destinatário (linha 11), e o servidor confirma ambas as requisições (linhas 10 e 12) com "250 OK". O cliente sinaliza o início do conteúdo da mensagem utilizando o comando "DATA" (linha 13), e o servidor usa o comando "354" (linha 14) para confirmar. Nas linhas 15 a 18 observamos a mensagem enviada pelo cliente, e na linha 19 a indicação de encerramento (com o ponto final). O servidor confirma então com "250 OK", o cliente encerra com "QUIT" e o servidor fecha a transmissão com "221 mx.google.com Service closing transmission channel".

O exemplo da Figura 8 ilustra uma característica fundamental do protocolo SMTP, que é sua suscetibilidade a observação e manipulação por um atacante situado no caminho entre cliente e servidor. Como o tráfego SMTP é todo em texto claro, um atacante no caminho poderá ver todas as mensagens, e quem está se comunicando. Esse atacante também poderá manipular o tráfego SMTP, incluindo, suprimindo ou modificando mensagens entre cliente e servidor. Isso se deve ao fato do protocolo SMTP ser um protocolo baseado em TCP e que não usa criptografia, o que o deixa vulnerável a um atacante no caminho.

Figura 8 – Exemplo de transação SMTP

```
S: 220 mx.google.com Simple Mail Transfer Service Ready
C: EHLO hotmail.com
S: 250-mx.google.com at your service
S: 250-SIZE 157286400
S: 250-8BITMIME
S: 250-STARTTLS
S: 250-ENHANCEDSTATUSCODES
S: 250 PIPELINING
C: MAIL FROM:<alice@gmail.com>
S: 250 OK
C: RCPT TO:<bob@hotmail.com>
S: 250 OK
C: DATA
S: 354
C: Subject: Encontro de quarta-feira
C: Ola Bob,
C: Tudo certo para nosso encontro de quarta?
C: Att, Alice
C: .
S: 250 OK
C: QUIT
S: 221 mx.google.com Service closing transmission channel
```

Fonte: O autor

2.4.3 STARTTLS

Quando utilizamos um navegador web, como Chrome ou Firefox, podemos verificar na URL o uso do HTTPS que garante a segurança na navegação, mas ao enviarmos um email possuímos o endereço para sinalizar o destino (`usuário@domínio`) e nenhuma forma de indicar que esse canal está protegido (ROSE et al., 2019). Como vimos anteriormente, o SMTP não é um protocolo fim a fim entre os usuários de origem e destino, mas sim um protocolo que envia mensagens de email através de saltos (MSA e MTAs). Isso implica um problema tanto para o remetente quanto para o receptor, pois não há como sinalizar que o envio da mensagem é segura e nem que qualquer dos saltos na transmissão é seguro (DUKHOVNI; HARDAKER, 2015b; ROSE et al., 2019). Uma extensão relacionada à segurança para SMTP é STARTTLS, definido na RFC 3207 (HOFFMAN, 2002). O STARTTLS permite adicionar autenticação e confidencialidade na comunicação entre agentes SMTP pelo estabelecimento de uma sessão *Transport Layer Security* (TLS) dentro da conexão SMTP. O STARTTLS foi muito bem aceito e implantado pelas grandes empresas como por exemplo Amazon, Facebook, Google, Microsoft e Yahoo (TUNG, 2016). Segundo (STALLINGS, 2016), se o cliente iniciar a conexão por meio de uma porta habilitada para TLS, o servidor poderá exibir uma mensagem indicando que a opção STARTTLS está disponível. O cliente pode então emitir o comando STARTTLS no fluxo de comando SMTP e as duas partes continuam a estabelecer uma conexão TLS segura.

Figura 9 – Uso do STARTTLS em SMTP

```
1  S: 220 mx.google.com Simple Mail Transfer Service Ready
2  C: EHLO hotmail.com
3  S: 250-mx.google.com at your service
4  S: 250-SIZE 157286400
5  S: 250-8BITMIME
6  S: 250-STARTTLS
7  S: 250-ENHANCEDSTATUSCODES
8  S: 250-STARTTLS
9  S: 250 PIPELINING
10 C: STARTTLS
11 S: 220 Go ahead
12 C: <inicia handshake TLS>
13 C e S: <estabelecem canal TLS>
```

Fonte: O autor

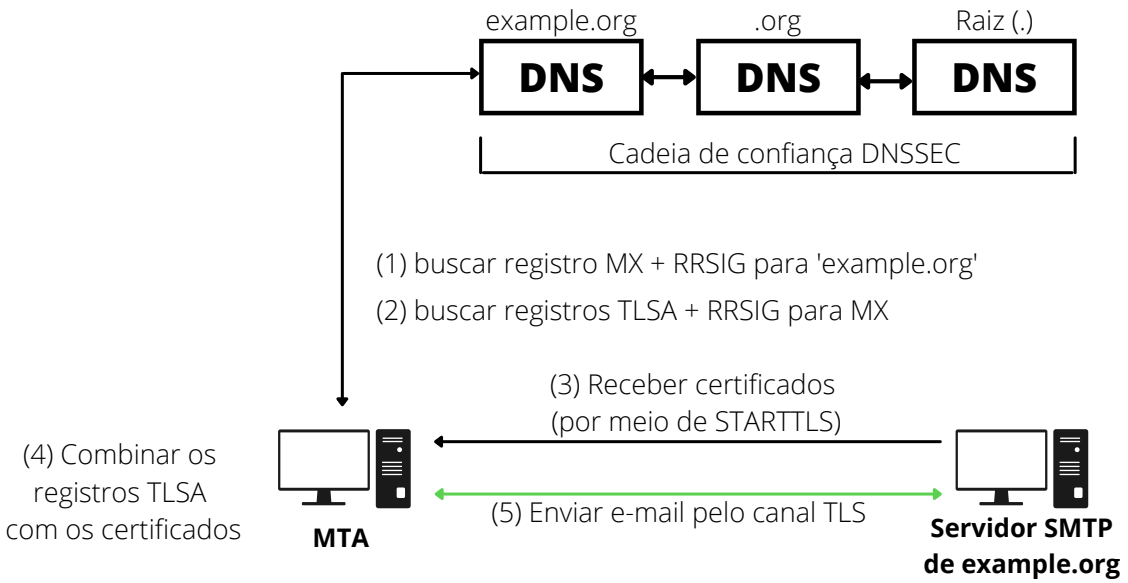
A Figura 9 ilustra o uso do STARTTLS no exemplo da Figura 8. Agora, o servidor SMTP anuncia com "250-STARTTLS" que oferece suporte a STARTTLS (linha 8). O cliente pode então enviar o comando STARTTLS (linha 10) e, após receber a confirmação do servidor com um código de resposta 220 (linha 11), iniciar o *handshake* TLS e estabelecer um canal seguro com o servidor (a partir da linha 12). Toda a identificação de remetente e destinatário e o conteúdo das mensagens (isto é, o tráfego a partir da linha 9 na Figura 8) será enviado de forma segura para o servidor.

A conexão inicial de um servidor de email para outro sempre começa sem criptografia, como ilustrado na Figura 9. Desta forma, um atacante que esteja no caminho entre cliente e servidor pode suprimir a mensagem de anúncio "250-STARTTLS" do servidor (linha 6) para fazer parecer que o TLS está indisponível; este ataque é chamado de ataque STRIPTLS (DUKHOVNI; HARDAKER, 2015b). Se um servidor de email não oferecer a opção do uso de STARTTLS durante o *handshake* SMTP (porque foi removido por um invasor), o transporte de email ocorrerá por meio de uma conexão não criptografada (BAATEN, 2019).

O DANE permite anunciar suporte para SMTP seguro por meio do registro TLSA. Isso informa aos clientes de conexão que eles devem exigir TLS, evitando assim ataques como o citado acima (STRIPTLS). Na Figura 10 é possível observar uma visão geral de como o DANE funciona em conjunto com DNSSEC e STARTTLS para o envio de um email para `alguem@example.org`. No passo (1), os registros MX para o domínio `example.org` são buscados no DNS. No passo (2), são buscados registros TLSA para um ou mais dos MX obtidos no passo (1); a presença desses registros sinaliza que o MX suporta TLS, e que o cliente deve encerrar a conexão caso não consiga estabelecer o canal TLS com esse servidor (DUKHOVNI; HARDAKER, 2015b). Em ambos os passos, a validação da cadeia de confiança DNSSEC garante a integridade e a autenticidade dos registros DNS recebidos, assim como da eventual inexistência de registros TLSA. No passo (3), o cliente SMTP envia o comando STARTTLS e recebe o certificado do servidor. No passo (4),

esse certificado é confrontado com os registros TLSA para fins de validação, especificamente utilizando o campo “dados de associação do certificado” do registro TLSA, que especifica o valor completo ou resumido de um certificado ou chave pública (DUKHOVNI; HARDAKER, 2015b). Se tudo estiver correto, o cliente finaliza o *handshake* TLS com o servidor. Por fim, no passo (5), o canal TLS é usado para enviar uma ou mais mensagens para `alguem@example.org`.

Figura 10 – Visão Geral do DANE



Fonte: Adaptado de (LEE et al., 2020)

Nessa seção foi apresentado diversos conceitos fundamentais e essenciais para o bom funcionamento do ecossistema DANE. Na próxima Seção iremos discutir sobre diversos trabalhos que possuem assuntos relacionados aos que discutimos tanto nessa Seção como nas Seções anteriores, tratando de DANE, STARTTLS, registro TLSA e DNSSEC.

2.5 TRABALHOS RELACIONADOS

Esta seção discute trabalhos na literatura relacionados com o tema deste TCC, concentrando-se em trabalhos que realizam medições sobre DANE e/ou STARTTLS.

Em (ZHU et al., 2015) é apresentado um estudo sistemático da implantação do DANE TLSA. Nesse estudo foi desenvolvida uma ferramenta que analisava ativamente todas as zonas assinadas em `.com` e `.net` em busca de registros TLSA, vale ressaltar que tal ferramenta não foi disponibilizada. Os dados foram coletados no segundo semestre de 2014. Como resultado, foi descoberto que o uso de DANE TLSA era precoce, isso porque na última medição (dezembro de 2014), dado um total de 485 mil zonas assinadas, foram encontrados apenas 997 registros TLSA. As medições também mostraram que cerca de 7-–13% dos registros TLSA eram inválidos (sem endereço IP, sem certificado, ou com divergência entre o registro TLSA e o certificado).

No estudo de Dongen (2018), ele investigou quais e quantas técnicas de segurança de email foram adotadas por organizações na Holanda. Uma lista de organizações holandesas com mais de 10 funcionários (no total 46.650 organizações únicas) foi criada, então foram definidos 19 parâmetros diferentes que foram verificados durante o experimento, incluindo verificar:

- Se o domínio é assinado por DNSSEC;
- Se o domínio está assinado com uma assinatura válida;
- Se os domínios para os quais os registros MX apontam são assinados por DNSSEC;
- Se os domínios para os quais os registros MX apontam estão assinados com uma assinatura válida;
- Se o servidor de recebimento de email oferece suporte à opção STARTTLS;
- Se o servidor de email de recebimento oferece suporte suficiente para uma versão TLS segura;
- Se o servidor de email oferece suporte a pacotes de criptografia suficientemente seguros;
- Uma cadeia de confiança válida deve ser publicada por autoridades de certificação confiáveis;
- Se o comprimento de bits da chave pública do certificado do servidor de email é suficientemente seguro;
- Se a impressão digital assinada do certificado do servidor de email foi criada com um algoritmo de hash seguro;
- Se o nome de domínio do servidor de recebimento de email (MX) corresponde ao nome de domínio no certificado.
- Se o servidor de email oferece suporte a compressão TLS;
- Se cada um dos domínios do servidor de correio fornece um registro TLSA para DANE;
- Se as impressões digitais DANE apresentadas pelos domínios do servidor de correio são válidas para os certificados do servidor de correio.

Foi usada uma ferramenta disponível em `internet.nl` para verificar se essas técnicas de segurança de email foram adotadas ou não. Como resultado, descobriu-se que as organizações adotaram em média 8,66 de um total de 19 parâmetros. Isso significa que os servidores de email de organizações holandesas adotaram menos de 50% das técnicas de segurança de email definidas pelo governo holandês. O setor de serviços públicos teve a pontuação mais alta, com uma pontuação média de 13,18. A pontuação mais alta está relacionada a políticas obrigatórias

para organizações governamentais porque muitas organizações governamentais estão presentes no setor de serviços públicos.

Sandelin (2017) realizou uma pesquisa quantitativa com os domínios de segundo nível suecos (isto é, sob o domínio `.se`) com o objetivo de estabelecer o nível de implantação de registros DANE TLSA. Os dados foram coletados em abril de 2017. Como resultado, obteve-se que o número de domínios assinados DNSSEC que foram validados com sucesso era grande (mais de 686 mil domínios), mas foram encontrados apenas 79 domínios com registros DANE (0,1% do total).

Szalachowski e Perrig (2017) realizaram um estudo com base na segurança do uso do DNS, ou seja, mostraram características das soluções, como por exemplo o DANE, e mediram a confiabilidade do DNS nessas aplicações. Foi observado que restringir outros RRs (especialmente relacionados a PKI, como TLSA) na verdade diminuirá a segurança dos usuários finais. Portanto, para permitir que as pessoas melhorem sua privacidade e segurança na Internet, os desenvolvedores devem considerar o suporte a aprimoramentos de segurança baseados em DNS.

Kambourakis, Gil e Sanchez (2020) trataram sobre os atuais padrões de segurança adotados na prática pelos provedores para proteger as comunicações entre seus servidores SMTP. Para isso, eles desenvolveram uma ferramenta chamada MECSA, que está disponível como um serviço de aplicação web para quem deseja avaliar o status de segurança de seu provedor de email em relação aos canais de comunicação de entrada e saída. Ao longo de 15 meses foram coletados alguns dados pela ferramenta MECSA, analisando um total de 3.236 provedores de email únicos, observou-se a taxa de adoção de extensões de segurança de email de última geração, incluindo STARTTLS, SPF, DKIM, DMARC e MTA-STS. Apos a aplicação da ferramenta, eles tiraram algumas conclusões nesse trabalho como por exemplo, que o STARTTLS trouxe resultados muito favoráveis quando utilizado e que o DNSSEC por ser um fator-chave na segurança de email afeta fortemente deixando o ecossistema de email muito mais seguro quando utilizado.

Tatang, Flume e Holz (2021) conduziram o primeiro estudo de medição longitudinal em larga escala sobre a adoção do MTA-STS (Mail Transfer Agent Strict Transport Security), uma alternativa ao DANE para sinalização segura do uso de TLS em servidores SMTP (MARGOLIS et al., 2018). Nesse artigo foi mostrado que o uso do MTA-STS é bastante baixo, ocorrem erros de configuração, e foi colocado que o MTA-STS provavelmente já está obsoleto devido ao uso do DANE. Porém eles não chegaram a fazer uma comparação entre o MTA-STS e o DANE, isso foi deixado como um trabalho futuro.

Holz et al. (2016) apresentaram o maior estudo até 2016 que investigava a segurança da infraestrutura de email e chat. Eles realizaram varreduras ativas em toda a Internet para determinar a quantidade de implantações seguras de serviços, e também fizeram um monitoramento passivo para averiguar em que grau os agentes de usuário escolhem mecanismos seguros para a sua comunicação. Segundo eles, os resultados obtidos foram uma mistura de descobertas positivas e negativas, porque embora os grandes provedores ofereçam boa segurança para seus usuários, a maior parte da comunicação é mal protegida em trânsito, com pontos fracos na configuração

criptográfica e especialmente na escolha dos mecanismos de autenticação. Além disso, foi descoberto que muitas configurações de cliente para servidor, especialmente para SMTP, não usavam credenciais válidas, isso significa que o email em trânsito pode muitas vezes ser entregue por meio de saltos não criptografados e não autenticados.

No artigo de Lee et al. (2020), que é a base para a realização deste trabalho de conclusão de curso, encontra-se um estudo de medição abrangente sobre o quão bem o padrão DANE e seus protocolos relevantes são implantados e gerenciados. Nesse estudo foram coletados dados para todos os domínios de segundo nível nos sob os domínios `.com`, `.net`, `.org`, `.nl` e `.se` durante um período de 24 meses para analisar a implantação e o gerenciamento do lado do servidor. Também foram analisados a implantação e o gerenciamento do lado do cliente, onde investigaram 29 provedores de serviços de email populares, quatro MTA populares e dez programas de software DNS. O estudo revelou que (1) a implantação do DANE é escassa, mas está aumentando, (2) mais de um terço de todos os registros TLSA não puderam ser validados devido a registros DNSSEC ausentes ou incorretos e (3) 14% dos certificados TLS apresentados por servidores de email estavam inconsistentes com os respectivos registros TLSA.

Este trabalho de conclusão de curso irá replicar a metodologia proposta por (LEE et al., 2020), realizando medições sobre o uso de DANE para proteção do serviço de email na Internet brasileira (`.br`), identificando a adoção do DANE em domínios brasileiros e analisando se o padrão está sendo usado corretamente. Também como um objetivo secundário será testado a reprodutibilidade desse trabalho acadêmico feito pelo (LEE et al., 2020)

2.6 CONSIDERAÇÕES DO CAPÍTULO

Este capítulo apresentou uma revisão de diversos conceitos que servem de base para este trabalho de conclusão de curso, incluindo aspectos do serviço de nomes (DNS, DNSSEC, e DANE) e email (SMTP e STARTTLS). É possível observar que esses vários protocolos precisam estar devidamente coordenados para garantir a proteção criptográfica das comunicações entre servidores de email usando canais TLS. Os trabalhos relacionados mostram que a adoção dos mecanismos de segurança vem se consolidando em algumas partes da Internet, havendo ainda uma lacuna no que tange a essa adoção no contexto brasileiro.

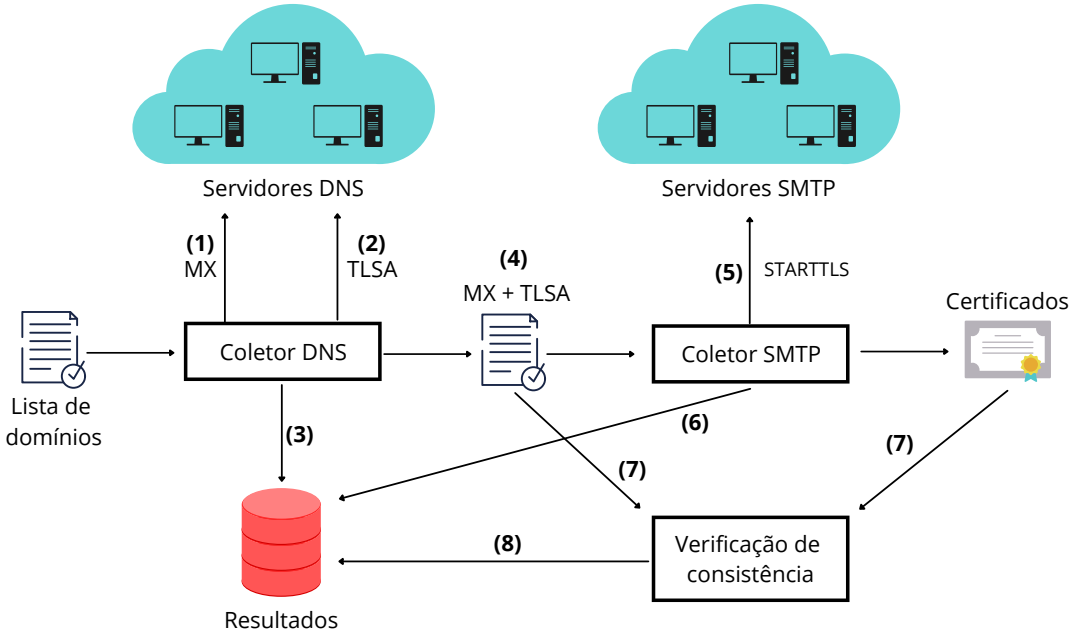
Este trabalho pretende preencher esta lacuna, replicando o estudo de Lee et al. (2020) com domínios brasileiros. O Capítulo 3 detalha melhor a proposta do trabalho, descrevendo as medições que serão realizadas e as métricas que serão analisadas.

3 PROPOSTA

Este trabalho de conclusão de curso visa investigar a adoção do ecossistema DANE e para proteção de email na Internet brasileira. Para isso, será reproduzido o estudo realizado por Lee et al. (2020), mas tendo como alvo domínios sob o `.br`. Pretende-se utilizar os artefatos disponibilizados pelos pesquisadores do estudo original na análise de dados, permitindo também avaliar se esses artefatos contribuem para a reprodutibilidade da metodologia.

A proposta deste trabalho está subdividida em duas etapas, uma delas é a coleta dos dados e a outra envolve toda a análise desses dados. Podemos observar de maneira mais ilustrativa essas duas fases na Figura 11.

Figura 11 – Coleta e análise dos dados



Fonte: Imagem criada pelo autor

O ponto de partida do processo é uma lista de domínios de interesse. Na fase de coleta de dados, observamos no passo (1) o coletor DNS buscando os registros MX para todos os domínios na lista. Na fase (2), o coletor DNS busca pelos registros TLSA para um ou mais MXs que foram obtidos no passo anterior. Como visto na Seção 2.4.3, o registro TLSA garante que o MX tenha suporte a TLS. Assim, no passo (4) é produzida uma lista dos MXs com registros TLSA correspondentes; esses servidores serão contactados para obter os certificados. Sendo assim, o coletor SMTP através do STARTTLS faz a conexão com os servidores SMTP (5) que devolve os certificados que serão verificados com os registros TLSA para serem validados.

Após a coleta de dados, será feita toda a análise dos dados. Algumas análises serão:

- No passo (3) os dados coletados sobre os MXs e os TLSAs serão analisados, assim vamos poder verificar quantos MX possuem TLSA e quantos não possuem.

- Após a coleta dos dados no passo (6), será possível verificar quais MXs possuem START-TLS e os que não possuem.
- Após a verificação de consistência dos certificados com os TLSAs, obtemos a última coleta de dados no passo (8), esses dados irão nos mostrar quantos domínios estabeleceram um canal TLS com o servidor, tal canal que é utilizado para o envio de mensagens de maneira segura e criptografada.
- Será possível verificar a porcentagem de todos os domínios (disponíveis na lista de domínios) que chegaram até a etapa final, ou seja, que possuem os registros MX e TLSA, que utiliza o comando STARTTLS e que tem seu certificado validado com o registro TLSA.
- Observar a porcentagem de conexões SMTP estabelecidas que falham ao iniciar conexões TLS, que normalmente, segundo Lee et al. (2020), essa falha se dá pela falta de registro DS.
- Analisaremos a porcentagem de registros TLSA que são DNSSEC inválidos devido a configurações DNSSEC erradas, como RRSIGs expirados e registros TLSA inconsistentes com os certificados.

Dentre as ferramentas descritas acima, será necessário desenvolver o coletor DNS, haja vista que um coletor SMTP e scripts para verificação de consistência são disponibilizadas por Lee et al. (2020). Para a implementação do coletor DNS, está-se avaliando a possibilidade de uso do `dns-crawler`[1], uma ferramenta de código aberto fornecida pelo projeto ADAM (Advanced DNS Analytics and Measurements)[2], vinculado ao NIC.CZ, que é a entidade responsável pelo domínio de primeiro nível `.cz` (República Checa).

Como já mencionado, o ponto de partida do processo de medição é a lista de domínios que serão analisados. Atualmente, não existe uma lista pública com todos os domínios brasileiros. Pretende-se elaborar a lista de domínios compilando listas de sites populares[3,4] e outros conjuntos de dados disponíveis publicamente[5,6].

3.1 CONSIDERAÇÕES DO CAPÍTULO

Este capítulo apresentou os principais aspectos da proposta do trabalho, abrangendo as etapas de coleta e de análise dos dados. Discutiu-se ainda o que precisará ser implementado e o

[1] `https://gitlab.nic.cz/adam/dns-crawler`
[2] `https://adam.nic.cz/en/`
[3] `https://tranco-list.eu/`
[4] `http://s3-us-west-1.amazonaws.com/umbrella-static/index.html`
[5] `https://ant.isi.edu/datasets/dns/index.html`
[6] `https://transparencyreport.google.com/safer-email/overview`

que poderá ser reusado, bem como possíveis estratégias para a construção da lista de domínios que serão investigados.

O próximo capítulo traz as conclusões do trabalho em seu atual estágio de desenvolvimento, e discute quais serão os próximos passos.

4 CONCLUSÃO

O uso somente do protocolo SMTP para o envio de emails não oferece garantia de segurança alguma, pois esse protocolo não possui mecanismos de criptografia. Para isso, a extensão STARTTLS permite que clientes e servidores estabeleçam um canal seguro TLS dentro de uma sessão SMTP. O STARTTLS oferece no inicio de toda conexão SMTP a opção do uso de um canal seguro TLS (ambos servidores devem suportar essa opção). Quando usado de forma isolada, o STARTTLS apresenta dois problemas. O primeiro é a vulnerabilidade a um ataque de *downgrade*, em que um atacante capaz de manipular o tráfego entre cliente e servidor suprime a mensagem que sinaliza o suporte a STARTTLS, forçando assim o uso de SMTP sem criptografia. O segundo problema está relacionado à validação do certificado TLS oferecido por um servidor SMTP. Por um lado, é frequente o uso de certificados cuja cadeia de confiança não tem como raiz uma das âncoras de confiança da PKI da web. Por outro lado, a PKI da web tem um grande número de âncoras de confiança, o que a torna vulnerável à emissão de certificados fraudulentos caso qualquer uma de centenas de autoridades certificadoras seja enganada ou tenha sua chave privada comprometida.

Uma solução para ambos os problemas é o uso do padrão DANE, que utiliza os registros TLSA no DNS (cuja integridade e autenticidade são garantidas através do DNSSEC) de forma a sinalizar de maneira segura um servidor de email que suporte o uso de STARTTLS. Os certificados publicados via DANE são emitidos pelo próprio servidor TLS, assim não dependem de uma autoridade certificadora.

Como não existe nenhum dado disponível sobre o uso do DANE para email na internet brasileira, este trabalho de conclusão de curso tem o objetivo de trazer tais informações para o domínio `.br` através da replicação do estudo já realizado por Lee et al. (2020). Um objetivo secundário é verificar a reprodutibilidade do estudo científico feito por Lee et al. (2020) através do reúso das ferramentas disponibilizadas pelos autores.

As próximas etapas a serem realizadas serão desenvolver o coletor DNS e testar as ferramentas disponibilizadas por (LEE et al., 2020), como já descrito no capítulo de proposta, em uma lista de domínios brasileiros. Tais ferramentas servirão tanto para a parte de coleta dos dados quanto também para a análise dos dados.

REFERÊNCIAS

ABLEY, J. et al. DNSSEC trust anchor publication for the root zone. **RFC 7958**, 2016. Disponível em: <https://datatracker.ietf.org/doc/html/rfc7958>. Acesso em: 5 jul. 2021. Citado na página 18.

ARENDS, R. et al. DNS security introduction and requirements. **IETF, RFC 4033**, 2005. Disponível em: <http://www.ietf.org/rfc/rfc4033.txt>. Citado 4 vezes nas páginas 7, 17, 18 e 19.

ARENDS, R. et al. Protocol modifications for the DNS security extensions. **IETF, RFC 4035**, 2005. Disponível em: <http://www.ietf.org/rfc/rfc4035.txt>. Citado na página 19.

ARENDS, R. et al. Resource records for the DNS security extensions. **IETF, RFC 4034**, 2005. Disponível em: <http://www.ietf.org/rfc/rfc4034.txt>. Citado na página 17.

ATKINS, D.; AUSTEIN, R. Threat analysis of the domain name system (DNS). **RFC 3833**, 2004. Disponível em: <https://datatracker.ietf.org/doc/html/rfc3833>. Acesso em: 5 jul. 2021. Citado na página 17.

BAATEN, Dennis. Better mail security with DANE for SMTP. **Apnic**, 2019. Disponível em: <https://blog.apnic.net/2019/11/20/better-mail-security-with-dane-for-smtp/>. Acesso em: 31 maio 2021. Citado 2 vezes nas páginas 7 e 26.

CERT.BR. Recomendações para evitar o abuso de servidores DNS recursivos abertos. **CERT – Centro de Estudos, Resposta e Tratamento de Incidentes de Segurança no Brasil**, 2013. Disponível em: <https://www.cert.br/docs/whitepapers/dns-recursivo-aberto/>. Acesso em: 15 jun. 2021. Citado na página 15.

CLARK, Jeremy; OORSCHOT, Paul C. van. SoK: SSL and HTTPS: Revisiting past challenges and evaluating certificate trust model enhancements. **IEEE Symposium on Security and Privacy**, p. 15, 2013. Disponível em: <http://www.css.csail.mit.edu/6.858/2020/readings/sok-ssl-https.pdf>. Citado 2 vezes nas páginas 7 e 20.

CLOUDFLARE. Como funciona o DNSSEC. 2021. Disponível em: <https://www.cloudflare.com/pt-br/dns/dnssec/how-dnssec-works/>. Acesso em: 16 jun. 2021. Citado na página 17.

COSTA, Daniel Gouveia. **DNS - Um Guia para Administradores de Redes**. Rio de Janeiro: Brasport, 2006. Citado na página 10.

CRISPIN, M. Internet message access protocol - version 4rev1. **RFC3501**, 2003. Disponível em: <https://datatracker.ietf.org/doc/html/rfc3501>. Acesso em: 03 ago. 2021. Citado na página 23.

CROCKER, D. Internet mail architecture. **RFC5598**, 2009. Disponível em: <https://datatracker.ietf.org/doc/html/rfc5598>. Acesso em: 03 ago. 2021. Citado 2 vezes nas páginas 22 e 23.

DÍAZ-SÁNCHEZ, Daniel. TLS/PKI challenges and certificate pinning techniques for iot and m2m secure communications. **IEEE**, 2019. Disponível em: <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=8704893>. Acesso em: 8 jul. 2021. Citado na página 20.

DONGEN, V. van. **Verifying email security techniques for Dutch organizations**. Dissertação (Mestrado) — University of Amsterdam, 2018. Citado na página 28.

DUKHOVNI, V. Opportunistic security: Some protection most of the time. **RFC 7435**, 2014. Disponível em: <https://datatracker.ietf.org/doc/html/rfc7435>. Citado na página 7.

DUKHOVNI, V.; HARDAKER, W. he DNS-based authentication of named entities (DANE) protocol: Updates and operational guidance. **RFC 7671**, 2015. Disponível em: <https://datatracker.ietf.org/doc/html/rfc7671>. Acesso em: 8 jul. 2021. Citado 3 vezes nas páginas 7, 20 e 22.

DUKHOVNI, V.; HARDAKER, W. SMTP security via opportunistic DNS-based authentication of named entities (DANE) transport layer security (TLS). **RFC 7672**, 2015. Disponível em: <https://datatracker.ietf.org/doc/html/rfc7672>. Acesso em: 8 jul. 2021. Citado 5 vezes nas páginas 7, 20, 25, 26 e 27.

ELZ, Robert; BUSH, Randy. Clarifications to the DNS specification. **RFC 2181**, 1997. Disponível em: <https://datatracker.ietf.org/doc/html/rfc2181>. Acesso em: 13 jun. 2021. Citado na página 13.

GELLENS, R.; KLENSIN, J. Message submission for mail. **RFC6409**, 2011. Disponível em: <https://datatracker.ietf.org/doc/html/rfc6409>. Acesso em: 03 ago. 2021. Citado na página 23.

GOLDBERG, S. et al. Stretching NSEC3 to the limit: Efficient zone enumeration attacks on NSEC3 variants. 2015. Disponível em: <https://www.cs.bu.edu/~goldbe/papers/nsec3attacks.pdf>. Acesso em: 5 jul. 2021. Citado na página 18.

HOFFMAN, P. SMTP service extension for secure SMTP over transport layer security. **RFC 3207**, 2002. Disponível em: <https://datatracker.ietf.org/doc/html/rfc3207>. Citado 2 vezes nas páginas 7 e 25.

HOFFMAN, P.; SCHLYTER, J. The DNS-based authentication of named entities (DANE) transport layer security (TLS) protocol: Tlsa. **RFC 6698**, 2012. Disponível em: <https://datatracker.ietf.org/doc/html/rfc6698>. Acesso em: 8 jul. 2021. Citado 4 vezes nas páginas 7, 20, 21 e 22.

HOLZ, Ralph et al. TLS in the wild: An internet-wide analysis of TLS-based protocols for electronic communication. **Technical University of Munich**, 2016. Disponível em: <https://www.net.in.tum.de/fileadmin/bibtex/publications/papers/2016-tls-ndss.pdf>. Acesso em: 03 ago. 2021. Citado na página 29.

KAMBOURAKIS, Georgios; GIL, Gerard Draper; SANCHEZ, Ignacio. What email servers can tell to johnny: An empirical study of provider-to-provider email security. **IEEE**, 2020. Disponível em: <https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=9139968>. Acesso em: 03 ago. 2021. Citado na página 29.

KLENSIN, J. Simple mail transfer protocol. **RFC 5321**, 2008. Disponível em: <https://datatracker.ietf.org/doc/html/rfc5321>. Citado 2 vezes nas páginas 7 e 23.

KNUBBEN, Bart. Overwiew of outbound DANE for smtp support. 2021. Disponível em: <https://github.com/baknu/DANE-for-SMTP/wiki/4.-Adoption-statistics>. Acesso em: 31 maio 2021. Citado na página 8.

KUROSE, J. **Redes de Computadores e a Internet**. 6th. ed. São Paulo: Pearson, 2010. Citado 2 vezes nas páginas 16 e 22.

LAURIE, B. et al. DNS security (DNSSEC) hashed authenticated denial of existence. **RFC 5155**, 2008. Disponível em: <https://datatracker.ietf.org/doc/html/rfc5155>. Acesso em: 5 jul. 2021. Citado na página 18.

LEE, Hyeonmin et al. A longitudinal and comprehensive study of the DANE ecosystem in email. In: **Usenix Security'20**. [s.n.], 2020. Disponível em: <https://taejoong.github.io/pubs/publications/lee-2020-dane.pdf>. Citado 7 vezes nas páginas 8, 9, 27, 30, 31, 32 e 34.

LIU, Cricket; ALBITZ, Paul. **DNS and BIND**. 5th. ed. United States of America: O'Reilly, 2006. Disponível em: <http://rauljesus.xyz/redes/dns/books/dnsAndBind5thEdition.pdf>. Acesso em: 7 jun. 2021. Citado 4 vezes nas páginas 10, 11, 12 e 14.

MARGOLIS, D. et al. SMTP MTA strict transport security (MTA-STS). **RFC 8461**, 2018. Disponível em: <https://datatracker.ietf.org/doc/html/rfc8461>. Acesso em: 8 jul. 2021. Citado na página 29.

MOCKAPETRIS, Paul. Domain names—concepts and facilities. **RFC 882**, 1983. Disponível em: <https://datatracker.ietf.org/doc/html/rfc882>. Acesso em: 7 jun. 2021. Citado na página 10.

MOCKAPETRIS, Paul. Domain names—implementation and specification. **RFC 883**, 1983. Disponível em: <https://datatracker.ietf.org/doc/html/rfc883>. Acesso em: 7 jun. 2021. Citado na página 10.

MOCKAPETRIS, Paul. Domain names - implementation and specification. **RFC 1035**, 1987. Disponível em: <http://tools.ietf.org/html/rfc1035>. Acesso em: 12 jun. 2021. Citado na página 16.

MOCKAPETRIS, Paul. Domain names—concepts and facilities. **RFC 1034**, 1987. Disponível em: <http://tools.ietf.org/html/rfc1034>. Acesso em: 12 jun. 2021. Citado 3 vezes nas páginas 11, 12 e 14.

MYERS, J.; ROSE, M. Post office protocol - version 3. **RFC1939**, 1996. Disponível em: <https://datatracker.ietf.org/doc/html/rfc1939>. Acesso em: 03 ago. 2021. Citado na página 23.

PENG, Roger. Reproducible research in computacional science. **American Association for the Advancement of Science**, 2011. Disponível em: <https://science.sciencemag.org/content/334/6060/1226/tab-pdf>. Citado na página 8.

POSTEL, Jonathan B. Simple mail transfer protocol. **RFC 821**, 1982. Disponível em: <https://datatracker.ietf.org/doc/html/rfc821>. Acesso em: 19 jul. 2021. Citado na página 23.

REGISTRO.BR. Introdução a DNS e DNSSEC. **NIC.BR - Núcleo de Informação e Coordenação do Ponto BR**, 2021. Disponível em: <https://ftp.registro.br/pub/doc/introducao-dns-dnssec.pdf>. Acesso em: 16 jun. 2021. Citado na página 17.

RESCORLA, E. The transport layer security (TLS) protocol version 1.3. **RFC 8446**, 2018. Disponível em: <https://datatracker.ietf.org/doc/html/rfc8446>. Citado 2 vezes nas páginas 7 e 20.

ROSE, S. et al. Trustworthy email. **NIST Special Publication 800-177**, 2019. Disponível em: <https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-177r1.pdf>. Acesso em: 19 jul. 2021. Citado 3 vezes nas páginas 22, 23 e 25.

SANDELIN, Rikard. Establishing dane TLSA deployment levels among swedish second level domains. **University os skovde**, 2017. Disponível em: <http://www.diva-portal.org/smash/get/diva2:1110505/FULLTEXT01.pdf>. Acesso em: 26 jun. 2021. Citado 3 vezes nas páginas 18, 21 e 29.

SCHUBA, C. L. Addressing weaknesses in the domain name system protocol. **University of Purdue**, West Lafayette, 1993. Disponível em: <https://www.cerias.purdue.edu/assets/pdf/bibtex_archive/94-05.pdf>. Acesso em: 7 jun. 2021. Citado na página 10.

STALLINGS, W. Comprehensive internet e-mail security. **The Internet Protocol Journal**, 2016. Disponível em: <http://ipj.dreamhosters.com/wp-content/uploads/issues/2016/ipj19-3.pdf>. Acesso em: 20 jul. 2021. Citado na página 25.

SZALACHOWSKI, Pawel; PERRIG, Adrian. Short paper: On deployment of DNS-based security enhancements. **Financial Cryptography and Data Security**, 2017. Disponível em: <http://fc17.ifca.ai/preproceedings/paper_64.pdf>. Acesso em: 03 ago. 2021. Citado na página 29.

TATANG, Dennis; FLUME, Robin; HOLZ, Thorsten. Extended abstract: A first large-scale analysis on usage of MTA-STS. **Ruhr-Universit¨at Bochum, Germany**, 2021. Disponível em: <https://www.ei.ruhr-uni-bochum.de/media/emma/veroeffentlichungen/2021/06/02/MTA-STS-DIMVA21.pdf>. Acesso em: 03 ago. 2021. Citado na página 29.

THO, Le Phuoc. DNSSEC policies in the wild. **Eindhoven University of Technology**, 2017. Disponível em: <https://pure.tue.nl/ws/portalfiles/portal/88385167>. Acesso em: 5 jul. 2021. Citado na página 17.

TUNG, L. Google, microsoft, yahoo: We want to stop email snooping by fixing these encryption flaws. **ZDnet**, 2016. Disponível em: <https://www.zdnet.com/article/google-microsoft-yahoo-we-want-to-stop-email-snooping-by-fixing-these-encryption-flaws/>. Acesso em: 20 jul. 2021. Citado na página 25.

VITEK, Jan; KALIBERA, Tomas. Repeatability, reproducibility and rigor in systems research. **Institute of Electrical and Eletronics Engineers**, 2011. Disponível em: <https://ieeexplore.ieee.org/abstract/document/6064509>. Acesso em: 25 maio 2021. Citado na página 8.

ZHU, Liang et al. Measuring DANE TLSA deployment. **University of Southern California**, 2015. Disponível em: <https://link.springer.com/content/pdf/10.1007/978-3-319-17172-2_15.pdf>. Acesso em: 03 ago. 2021. Citado na página 27.