



Understanding Network Applications

Network Applications and Endpoint Security

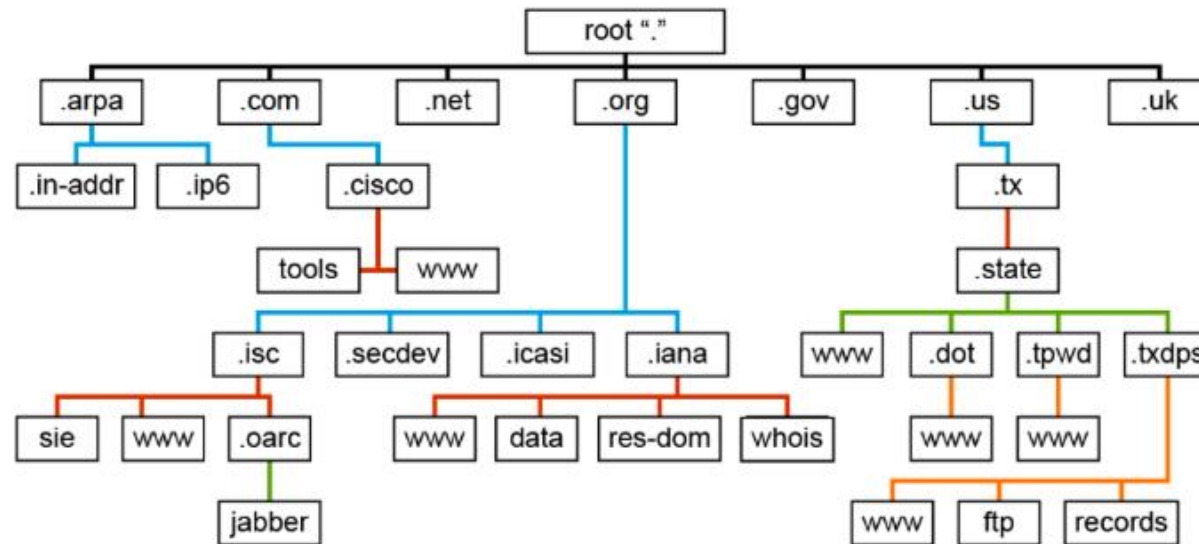
Ahmed Sultan
Senior Security Engineer
ahmedsultan.me/about

DNS Operations

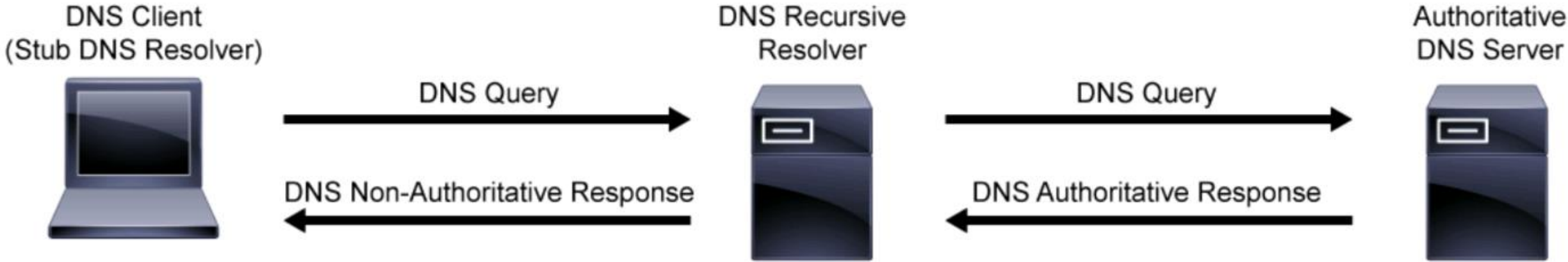
- DNS provides mappings between the host names and IP addresses, or other mappings
- The client side of DNS is called a DNS resolver
- DNS is a critical protocol for the network operations but weaknesses allow it to be exploited
- Primarily uses UDP port 53 for DNS queries and responses
- TCP port 53 is used when the DNS response data size exceeds 512 bytes, or for tasks such as zone transfers

DNS Distributed Database

- DNS is a globally distributed, scalable, hierarchical, and dynamic database
- The DNS database is composed of a hierarchical domain name space that contains a tree-like data structure of linked domain names (nodes)
- The topmost level of the DNS hierarchy is represented by the "dot" (.)



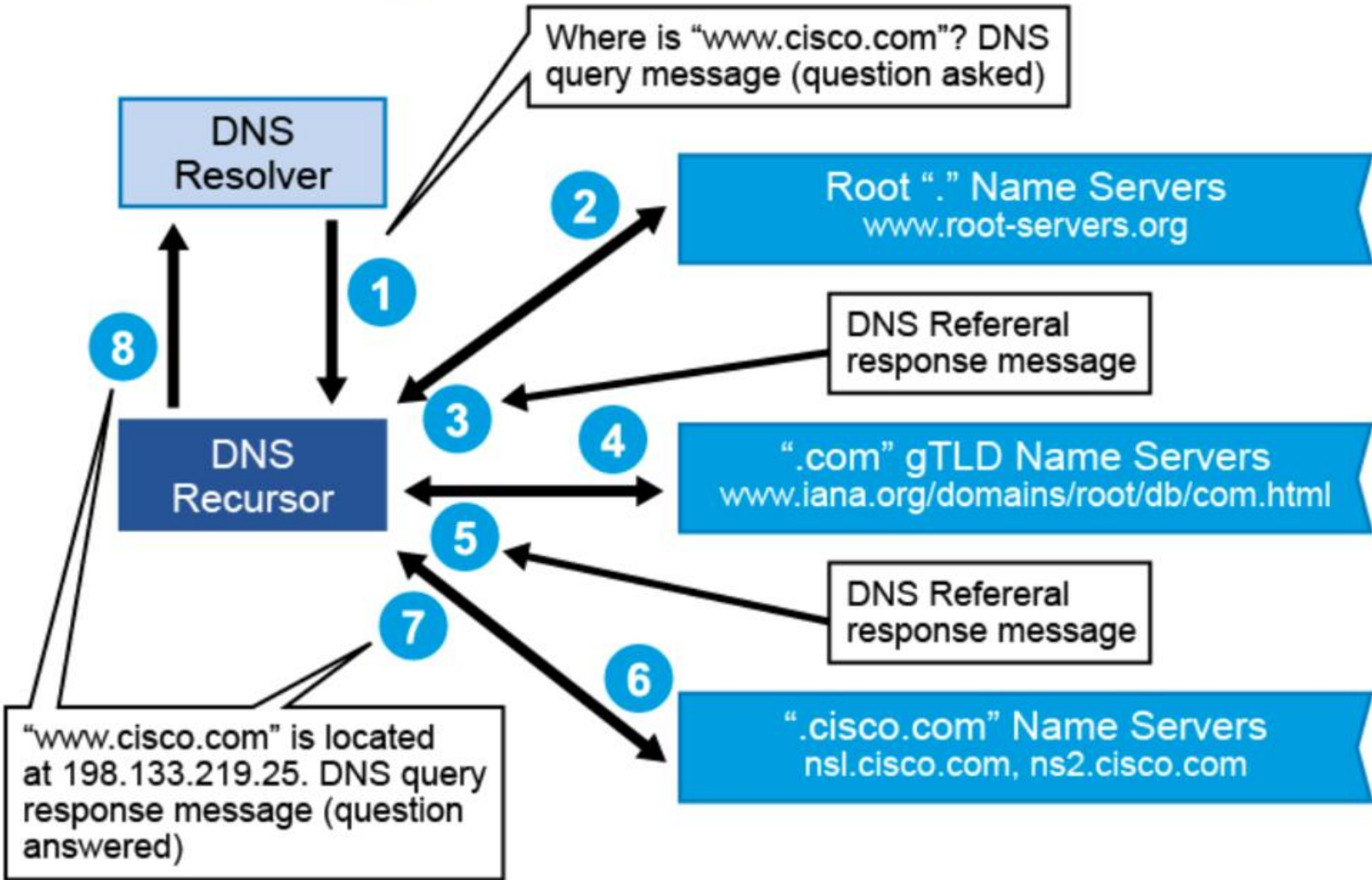
DNS in Action



DNS RR Types

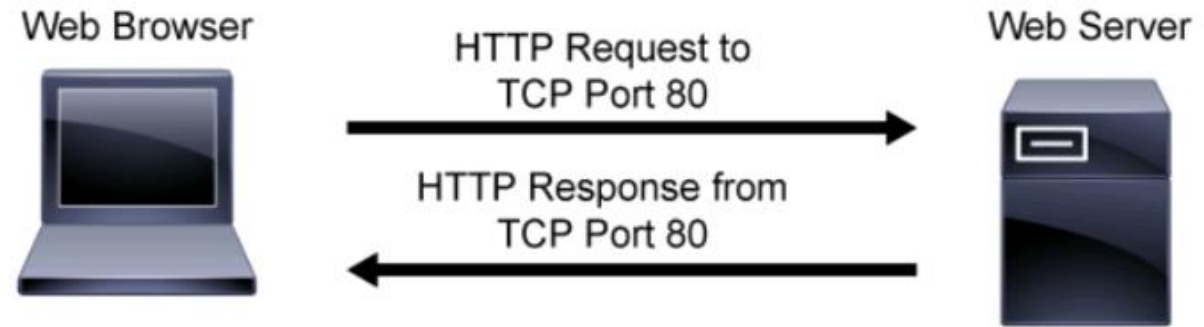
- **A record:** Used to map host names to the IPv4 address of the host. Multiple IP addresses can correspond to a single host name, and multiple host names, each of which maps to the same IP address.
- **AAAA record:** AAAA is used to map hostnames to the IPv6 address of the host.
- **MX record:** MX maps a domain name to a list of mail servers for that domain.
- **PTR record:** A PTR points to a canonical name.
- **NS record:** An NS record identifies the DNS servers that are responsible (authoritative) for a zone.
- **CNAME record:** A CNAME record is used to specify that a domain name is an alias for another domain name.
- **TXT record:** A TXT record is used to associate any arbitrary text with a hostname.
- **SOA record:** Each zone contains an SOA record. The SOA record identifies the name server that is the best source of information for the data within the zone.

Recursive DNS Query



HTTP Operations

- HTTP is a client/server protocol where the web browser is the client and the web server is the server.
- HTTP is a stateless application layer protocol.
- Default port for HTTP is TCP port 80, but other ports can be used.



URI and URL

http://www.example.cisco.com:80/video?docid=96673783583808&hl=en#00h01m15s

- **Protocol:** http (can also be https, ftp, and so on)
- **Host:** www.example.cisco.com
 - **Host (or Prefix)** = www. **Subdomain** = example.cisco.com. **Domain** = cisco.com. **Top-Level Domain** = .com.
- **Port:** If the port is not specified, port 80 is assumed.
- **Path:** /video. Path typically refers to a file or location on the web server. You can think of a path as a directory structure.

URI and URL (Cont.)

`http://www.example.cisco.com:80/video?docid=96673783583808&hl=en#00h01m15s`

- **Parameters:** `?docid=96673783583808&hl=en`. This example reference a specific video file in the path. The `hl=en` parameter specify the language, for example, setting the video subtitle to English.
 - URL parameters are also referred to as “query strings,” which contain extra information in the form of key-value pairs called parameters. Parameters start with a question mark (?) and are separated with an ampersand (&).
 - **Fragment or named anchor:** `#00h01m15s`. Typically the fragment is used to refer to an internal section within a web document.

HTTP Request Methods

Common HTTP request methods include GET, HEAD, POST, PUT, and DELETE

- The GET method retrieves data from the specified resource.
- The HEAD method asks for a response identical to that of a GET request, but without the response body
- The POST method creates data on the specified resource.
- The PUT method request is used to update data on the specified resource.
- The DELETE method deletes the specified resource.

HTTP Request and Response Packets Capture Example

The image shows a Wireshark packet capture window titled 'lab4-ssh.pcap [Wireshark 1.10.1 (SVN Rev 50926 from /trunk-1.10)]'. The filter bar shows 'tcp.stream eq 4'. The packet list shows three packets: a SYN packet (No. 64), an ACK packet (No. 65), and a GET packet (No. 66). The packet details pane shows the 'Follow TCP Stream' window for the GET packet. The stream content shows the following HTTP request and response:

```
GET / HTTP/1.1
User-Agent: wget/1.13.4 (linux-gnu)
Accept: */*
Host: 10.3.1.200
Connection: Keep-Alive

HTTP/1.1 200 OK
Date: Mon, 12 Oct 2015 14:24:57 GMT
Server: Apache/2.2.22 (Ubuntu)
Last-Modified: Fri, 15 Mar 2013 14:35:26 GMT
Etag: "45a28-b1-4d7f78cd62147"
Accept-Ranges: bytes
Content-Length: 177
Vary: Accept-Encoding
Keep-Alive: timeout=5, max=100
Connection: Keep-Alive
Content-Type: text/html

<html><body><h1>It works!</h1>
<p>This is the default web page for this server.</p>
<p>The web server software is running but no content has been added, yet.</p>
</body></html>
```

The packet bytes pane at the bottom shows the raw data of the captured packets.

HTTP Status Codes

Common status codes include the following:

- **100 = Continue:** The server has received the request headers and the client should proceed to send the request body (in the case of a request for which a body needs to be sent; for example, a POST request).
- **200 = OK:** The processing of the request that was sent by the client was successful.
- **301 = Moved Permanently:** The resource has permanently moved to a different URI.
- **302 = Found:** The requested resource resides temporarily under a different URI.
- **307 = Temporarily Moved:** The request should be repeated with another URI; however, future requests should still use the original URI.

HTTP Status Codes (Cont.)

Common status codes include the following:

- **401 = Unauthorized (Authentication Required):** The request first requires authentication with the server.
- **403 = Forbidden:** Access is denied.
- **404 = Not Found:** The server cannot find the requested URI.
- **407 = Proxy Authentication Required:** The request first requires authentication with the proxy.
- **500 = Internal Server Error:** This generic web server error message is given when an unexpected condition is encountered and no more specific message is suitable.

HTTP Cookies

- An HTTP cookie is a small piece of data that is sent from the web server and stored in the user's web browser.
- Cookies are used by the web server to remember stateful information.
- Can also be used to remember arbitrary pieces of information that were previously entered by the user in form fields, such as name and address.
- Cookies are passed between the web server and web browser using the **Set-Cookie** HTTP header field in the HTTP response, and the **Cookie** HTTP header in the HTTP request
- The **sessionToken** cookie is a piece of data that can be used by the web server to identify a particular session.

HTTPS Operations

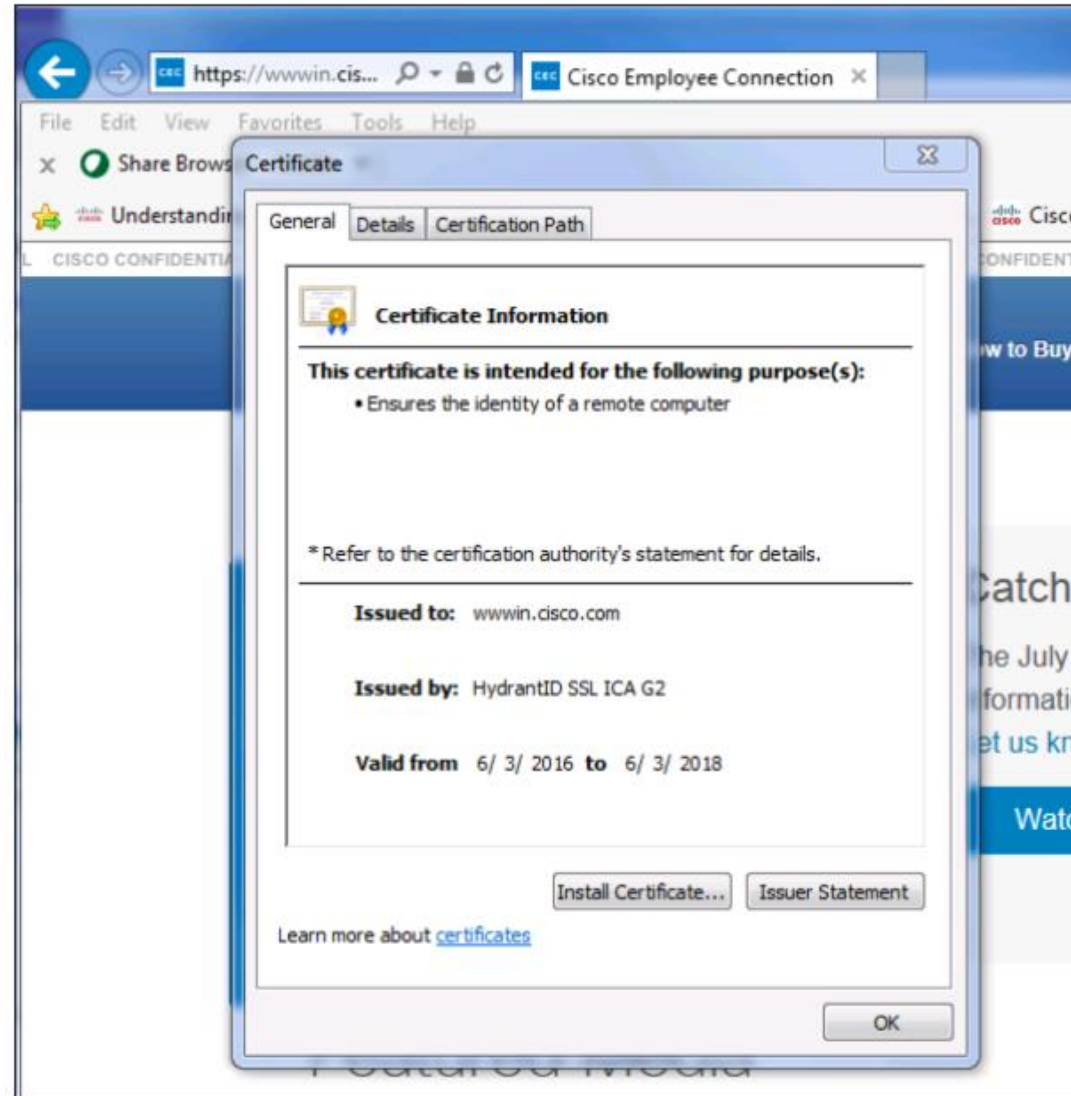
- HTTPS is a combination of HTTP and TLS or its predecessor, SSL.
 - HTTP runs on top of the TLS or SSL protocol.
- HTTPS should be used instead of HTTP whenever private data is being transmitted.
- TLS or SSL is used by HTTP to establish an encrypted connection to an authenticated peer over an untrusted network.

HTTPS Basic Operations

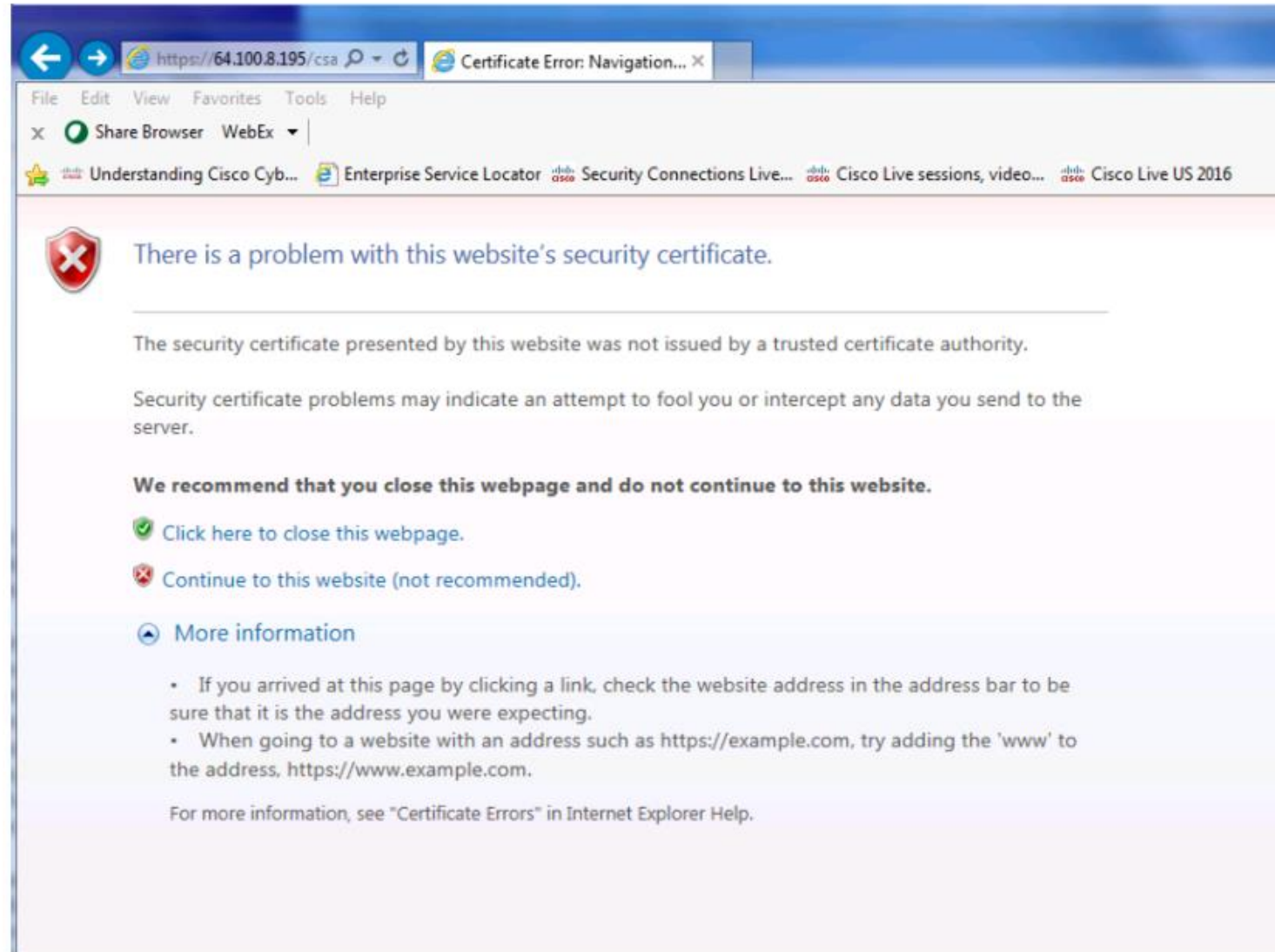
HTTPS basic operations include the following:

- HTTPS URLs begin with `https://` and use TCP port 443 by default.
- The TLS or SSL connection between a client and server is set up by the TLS or SSL handshake. Once the TLS or SSL handshake is established, both parties use the agreed cryptographic algorithms to securely send messages to each other.
- HTTPS provides authentication of the web server. The web server's digital certificate allows the browser to identify the web server.
- HTTPS can also provide mutual authentication. If client authentication is also required, the web server can also authenticate the client using the client's digital certificate.
- HTTPS provides HTTP headers and HTTP data traffic encryption between the client and the web server.

Web Server Digital Certificate: Valid Certificate Example

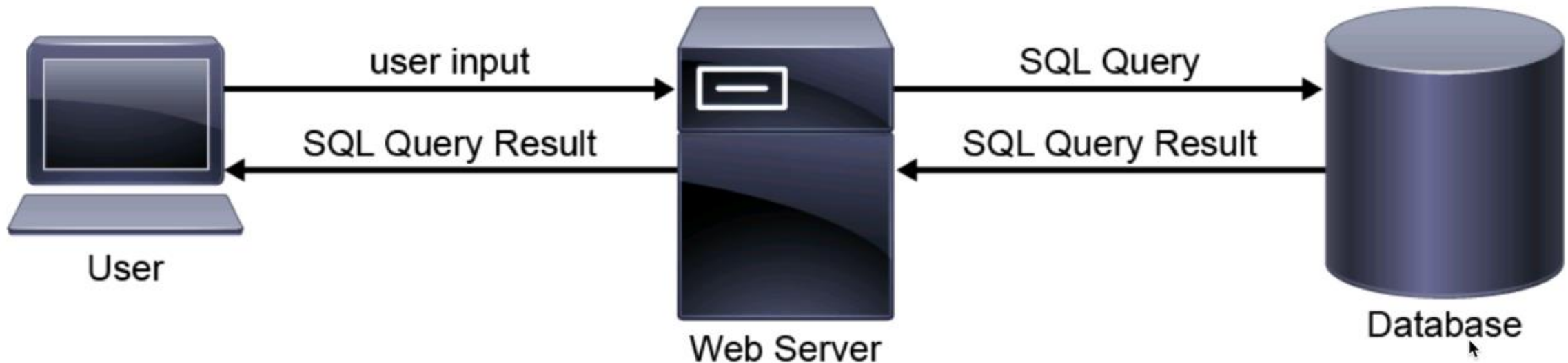


Web Server Digital Certificate: Untrusted Certificate Example



SQL Operations

- SQL is used to query, operate, and administer relational database management systems.
- An analyst should understand how to recognize SQL based attacks such as the SQL injection attack.
 - A SQL injection attack involves the alteration of SQL statements that are used within a web application.



SQL Functions

SQL functions include the following:

- Create databases and tables. The data in a database is stored in the tables. The table is a collection of related data entries and it consists of columns and rows. Columns contain the column name, data type, and any other attributes for the column. Rows contain the records or data for the columns.
- Define the data in the database and manipulate that data.
- Access the data in the database.
- Set the database permissions.

SQL Commands

The following SQL commands are grouped according to the attacker's goals:

- Exfiltrating data
 - **SELECT [fields] FROM [table] [...]**
- Modifying data
 - **UPDATE [table] SET [field] = [value] WHERE [condition]**
 - **INSERT INTO [table] VALUES [...]**
 - **TRUNCATE TABLE [table]**
- Modifying database structure
 - **DROP TABLE [table]**
 - **ALTER TABLE [table] [...]**
 - **DROP DATABASE**

SMTP

Two of the major threats to an organization's email system are:

- A flood of unsolicited and unwanted email, called spam, which wastes employee time through sheer volume and uses valuable resources like bandwidth and storage
- Malicious email, which comes in two basic forms of attacks: embedded attacks and targeted attacks
 - Embedded attacks come in the form of viruses and malware that perform actions on the end device when clicked.
 - Targeted attacks might direct employees to inadvertently browse malicious websites that distribute malware to computer endpoints and can mislead employees into releasing sensitive.
 - Targeted attacks are also known as directed attacks, or phishing attacks.

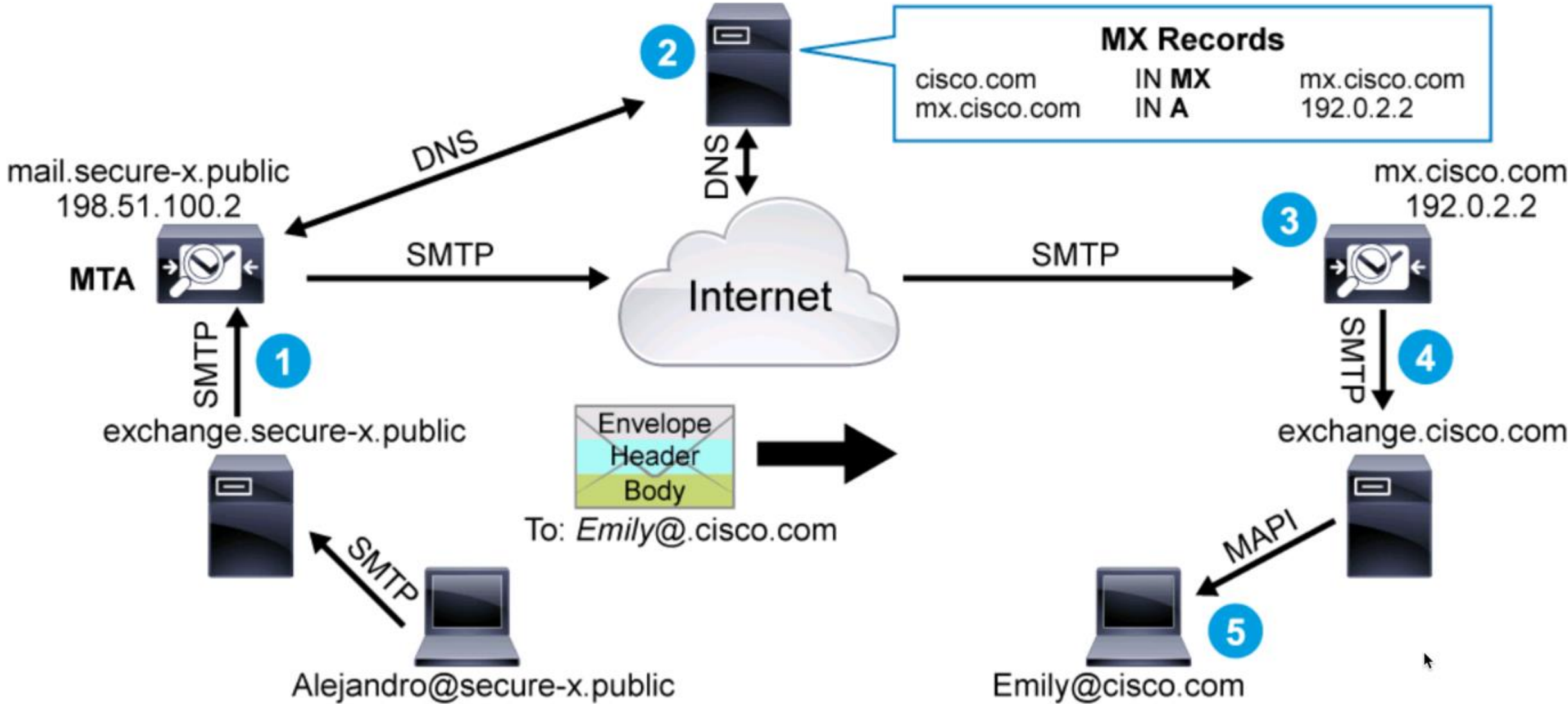
SMTP Terminology

- **MTA:** Also called SMTP daemon: Computer program or software agent that transfers electronic mail messages from one computer to another.
- **DNS MX record:** Mail exchanger record, type of resource record that specifies the mail server (MTA) responsible for accepting email for that domain.
- **DNS A record:** Used to locate the IP address of the MTA specified by the MX record.
- **Groupware server:** A server that accepts, forwards, delivers, and stores messages on behalf of users.
- **SMTP client:** Initiates connection request to an SMTP server.
- **SMTP server:** Will receive the connection request from the SMTP client.
- **Mail user agent:** The MUA is a software client application like Outlook that accesses a groupware server.

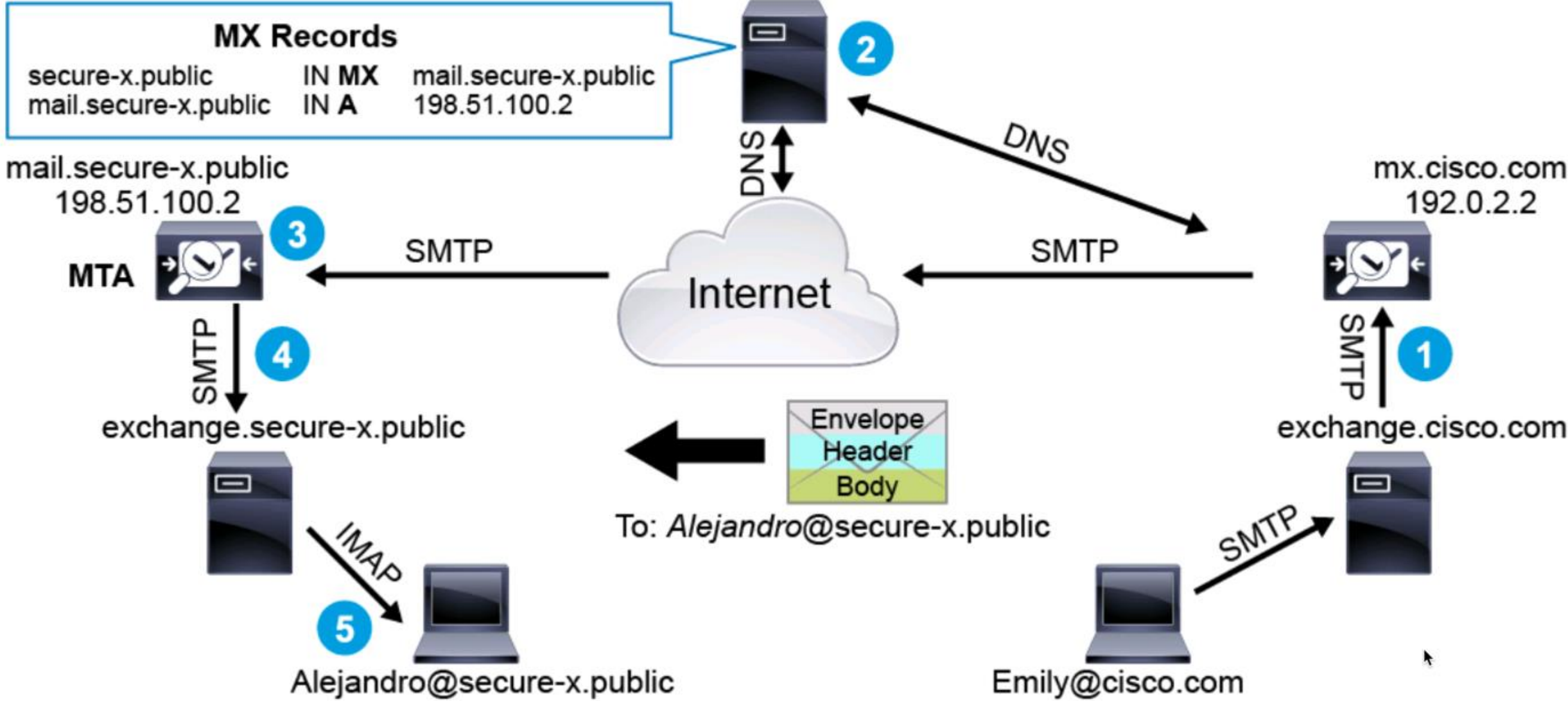
SMTP Terminology (Cont.)

- **POP:** Application-layer protocol that is used by the MUA to retrieve email from a mail server. TCP port 110.
- **IMAP:** Application-layer protocol that is used by the MUA to retrieve email from a mail server. TCP port 143.
- **MAPI:** Application-layer protocol that is used by the MUA to retrieve email from a mail server. Primarily associated with Microsoft Exchange and Microsoft Outlook.

SMTP Flow



SMTP Flow (Cont.)



SMTP Conversation

mail.secure-x.public
198.51.100.2

Envelope

```
<< 220 mx.cisco.com ESMTP
>> HELO mail.secure-x.public
<< 250 mx.cisco.com
>> MAIL FROM: <Alejandro@secure-x.public>
<< 250 sender <Alejandro@secure-x.public> ok
>> RCPT TO: <Emily@cisco.com>
<< 250 recipient <Emily@cisco.com> ok
```

Headers

```
>> DATA
<< 354 go ahead
>> From: Alejandro Martinez<Alejandro@secure-x.public>
>> To: Emily Chang<Emily@cisco.com>
>> Subject: Going to the Cyber security training?
>> Date: Wed, 8 August 2016 20:57:13 -0700
```

Body

```
>> I am so excited to be in the Cyber security training
>> Alejandro
<< 250 ok
>> QUIT
<< 221 mx.cisco.com
```

mx.cisco.com
192.0.2.2