



Hands-On with TLS and Certificates





Michael Smith

Field CTO – Vercara

- Field CTO for Vercara, an Infrastructure and Cybersecurity Cloud Provider (DNS, DDoS Mitigation, WAF, CDN, Bot Management)
- Former Akamai Security CTO for Asia-Pacific, Europe, Media, and Carrier
- Founder and Former Director of Akamai's Customer Security Incident Response Team
- 20+ years of web server, DNS, and infrastructure administration
- Part-time programmer
- CISSP-ISSEP and CISM certifications



Agenda

Section 1	Theory: What are all These Cryptography Terms? Exercise: Exploring TLS and Certificates on Popular Websites Theory: The CA Hierarchy and the Browser CA Keychain
Section 2	Theory: The Certificate Creation Process Exercise: Creating Your Own CA Exercise: Building a Webserver with TLS
Section 3	Exercise: Using SSL Labs
Section 4	Theory: LetsEncrypt Exercise: Deploying Certificates to Non-Web Services Theory: Cloud Service Providers and Certificates





Theory: What are all These
Cryptography Terms?

Session 1

Exercise: Exploring TLS and
Certificates on Popular Websites

Theory: The CA Hierarchy and the
Browser CA Keychain

Why Study TLS and PKI?

- TLS and PKI tie into web defense
- This is the easiest way in IT to make mistakes
- Browsers use TLS by default now
- Vulnerability scanners love to find TLS problems
- Good hands-on practice with implementing a cryptographic system from scratch
- Acronyms are fun!!!!



Some of my “Claims to Fame”

- My job at Akamai for 3+ years was to explain certificate management to CDN customers
- I’ve worked on several SSL protocol errors: Heartbleed, POODLE, Logjam, etc
- I explain TLS and PKI concepts at least monthly at Vercara
- I used OpenSSL command-line tools to create the digital signature standard for your plane tickets



2 Types of Ciphers

- Symmetric
 - Both sides of the conversation have the same key
 - Faster
 - Uses less CPU and RAM
 - Problems with key distribution and replacement
 - Examples: DES, 3DES, AES
- Asymmetric
 - Uses a public key and private key
 - Slower
 - Lots of CPU usage
 - Easier key distribution and replacement
 - Examples: RSA, ECC



Hashing Algorithms

- A one-way algorithm that “adds up the bits” in a non-reversible way
- Used as a one-way password store (/etc/shadow), identifier (malware samples), or an integrity control (ISO file downloads)
- Used in digital signatures
- Examples: MD5, SHA, SHA-2, SHA-256



Message Authentication Code

- Signature on a message
- Requires asymmetric encryption and a hash algorithm
- Provides integrity and non-repudiation
- TLS uses HMAC: keyed-hash message authentication code or hash-based message authentication code
- Message \Rightarrow hash it \Rightarrow encrypt the hash with your private key



Key Exchange in TLS

- We want to do a “handshake” to create a symmetric key to encrypt our conversation
- Called “Diffie-Hellman” key exchange
- There are a ton of variants:
 - Elliptic-curve D-H
 - Anonymous D-H
 - Fixed D-H
 - Ephemeral D-H
- Perfect Forward Secrecy (PFS) is where we create a new key for each session instead of reusing a previous key



X509 Certificates

- How you identify a server but sometimes a client
- Has a “signature chain” back to a “trusted” Certificate Authority
- Private key plus context in a wrapper:
 - Organization name and address (O)
 - Issued and expiration dates
 - Signature chain
 - Hostname (CN) and alternate names (SAN)
 - Serial number



X509 Certificates

<Demo>



The Evolution of SSL/TLS

- SSL V1
- SSL V2
- SSL V3
- TLS 1.0
- TLS 1.1
- TLS 1.2
- TLS 1.3



Other Weirdnesses

- Wildcard v/s Subject Alternate Names (SAN)
- Server Name Indicator (SNI)
- HTTP Strict Transport Security (HSTS)
- Certificate Revocation List (CRL)
- Mutual TLS (MTLS)



Session 1

Theory: What are all These
Cryptography Terms?

Exercise: Exploring TLS and
Certificates on Popular Websites

Theory: The CA Hierarchy and the
Browser CA Keychain



Let's Explore Certificates

- CN
 - SAN
 - Wildcards
 - CA
 - Intermediate CA
 - Expiration
 - Public Key
- Look at These:
 - disney.com
 - amazon.com
 - whatsapp.com
 - facebook.com
 - cloudflare.com
 - akamai.com





Theory: What are all These
Cryptography Terms?

Session 1

Exercise: Exploring TLS and
Certificates on Popular Websites

Theory: The CA Hierarchy and the
Browser CA Keychain

Certificate Authorities

- CAs pay to be in your browser store
- CAs have a ton of weird security requirements
 - Offline store in a safe
 - 2-person control to sign certificates
 - Ultra-long expiration dates
- CAs use an online intermediate CA to sign certificates
- Intermediate CAs can be revoked without publishing them in the browser



Certificate Authorities

- CAs create intermediate CA
- Intermediate CAs might have additional intermediate CAs
- Some companies get intermediate CAs for internal use
- Intermediate CAs sign certificates



Certificate Authorities

<Demo>





Theory: The Certificate Creation Process

Session 2

Exercise: Creating Your Own CA

Exercise: Building a Webserver with TLS

Creating a Certificate

Create a unique private key

Create a certificate signing request (CSR)

Submit the CSR to the CA

The CA signs the CSR with their intermediate CA

Import the signed CSR to make the certificate

Deploy the certificate and private key to the webserver



Generating a Certificate

<Whiteboard>





Theory: The Certificate Creation Process

Session 2

Exercise: Creating Your Own CA

Exercise: Building a Webserver with TLS

DIY Certificate Authority

Create a private key for your CA

Create a certificate from the CA private key

Create a private key for your certificate

Create a certificate for your webserver





Theory: The Certificate Creation Process

Session 1

Exercise: Creating Your Own CA

Exercise: Building a Webserver with
TLS

Deploying Certificate to Our Own Server

<Demo>

Session 3

Exercise: Using SSL Labs



Qualys SSL Labs

<Demo>





Theory: LetsEncrypt

Session 4

Exercise: Deploying Certificates to
Non-Web Services

Theory: Cloud Service Providers and
Certificates

LetsEncrypt

Free Public CA

Issues certificates for 90 days

Multiple methods of verification

- Object on the website

- DNS TXT record



LetsEncrypt

<Demo>



Session 4

Theory: LetsEncrypt

Exercise: Deploying Certificates to
Non-Web Services

Theory: Cloud Service Providers and
Certificates



TLS and Other Services

<Demo>





Theory: LetsEncrypt

Session 4 Exercise: Deploying Certificates to
Non-Web Services

Theory: Cloud Service Providers and
Certificates

Cloud Service Providers and CAs

AWS: Certificate Manager

Azure: Key Vault

GCP: Certificate Authority Service





Thank You All