The New Key Management:
Unlocking the Safeguards of Keeping Keys Private

Jono Bergquist
Solutions Engineering Lead - APJ
CloudFlare
Outline

- Why application-level TLS is important
- Key management is the hardest part of TLS
- How to use trusted computing for cryptography
- Solving TLS key management with TPMs
The perimeter is porous
Traditional Network Security Topology

Internet VPN

Regional Office
Regional Office
Head-office
Remote / roaming users
Traditional Network Security Topology

- Multiple internal services
  - Databases with customer data
  - Employee portals
- Cross-datacenter communication across Internet via **VPN**
  - All or nothing access
The perimeter is porous - VULCANDEATHGRIP
Traditional network topology

- VPN compromise makes application-to-application data readable
Web Application Security Topology
Edge Network
Mobile network
The modern corporate network

- Components
  - Website hosted on a SaaS/IaaS platform
  - Core business services
  - Loosely affiliated group of services hosted by third parties
The modern corporate network

- Access control
  - Third-party services
    - Federated identity (SAML, OAuth, etc.)
    - Single sign-on
  - Service-to-service authentication
    - Implicit via VPN
    - Token-based
Examples of application-to-application data

- Data breaches
  - User passwords
  - Customer data
  - HR Data
  - Customer lists
  - Proprietary intellectual property

- All from applications inside the network
The modern corporate network

- The perimeter is fuzzily defined
- Move security to a higher level in the stack?
Application-layer Encryption
Encryption

- Corporate data should be encrypted
Encryption

- ...at rest
- ...in transit
- ...with authentication
Layer 3 Encryption

- IPsec tunnel/VPN
  - Expensive hardware
  - Does not scale to edge networks
  - Trust everyone
Layer 5/6 Encryption

- Kerberos
  - Web applications do not use it
- Transport Layer Security
  - Widely supported among a range of applications
Transport Layer Security (TLS)

- The protocol formerly known as SSL
- Provides server-to-server encryption
- Authentication via certificate validation

Advantages
- Cheap in software on modern processors (AES-NI)
- Widely supported in service oriented software
Transport Layer Security (TLS)

- Challenges for application-to-application TLS
  - Building a system of trust
  - Key management
Building trust in applications
TLS without certificate validation

- Traditional man-in-the-middle attack
Trust Models for TLS

- Public Key Infrastructure model
- Each application has:
  - Public X.509 certificate
  - Corresponding private key
X.509 Public Key Infrastructure

The anatomy of a certificate

- Common Name and Organization
- DNS name
- Dates valid
- SSL Certificate
- Public Key
- Issuer Signature
- Issuer Name and Organization
Trust Models for TLS

- Session key used to encrypt connection
- Private key used to
  - Prove ownership of certificate
  - Authenticate session establishment
- Validate certificates with a chain of trust
Certificate chain of trust

- CloudFlare Offline Root
- CloudFlare Online Intermediate Root
- Application Certificate
- Offline private key
- Private key
- Private key
PKI-enabled applications

- Database access
- Business services
- Mobile applications
Private PKI

- Run your own internal Certificate Authority
- Generate keys locally on endpoints
- Use internal CA to create certificates
Different CAs for different domains

Certificate authority

API server CA

Database server CA

API server

Database server
Service to service communication
With TLS mutual authentication

API server
- Trusted CA
- Key share
- Session key

Database server
- Trusted CA
- Key share
- Session key

Cloudflare
Tools

- OpenSSL
- CFSSL
- CloudFlare’s open source CA software
- pki.io
- EJBCA
- Commercial options
Advantages

- Application data is encrypted in transit
- Requests are authenticated
- VPN failure is no longer catastrophic
The bootstrap problem

- Enrolling new servers
- Authenticating requests for certificates
Dangers

- Keys live in memory and on disk
- Can be stolen and applications impersonated
Trusting trusted computing
Protecting keys on servers

- Keep keys in hardware instead of software
- Each machine needs its own hardware
  - HSMs are prohibitively expensive
  - TPMs fit the bill ($15-$30 each)
Trusted Platform Module
Trusted Platform Module

- Most commonly used for Windows trusted boot

- List of features of TPM 1.2
  - Measured Boot
  - Random number generation
  - RSA 2048 private keys
Machine provisioning

Provisioning DB

TPM Public Key

TPM Public Key

TPM

TPM
Certificate issuance
Benefits

- Keys do not live in software
  - Safe from memory access (Heartbleed, DMA)
  - Safe from theft (TPM locked)
  - Safe from impersonation
Drawbacks

- Not all software supports TPM crypto
- It is sloooow
Simple guide
How to set up secure application transport

- Create your own CA on a trusted machine or HSM
- Create a key on your device TPM
- Use TPM to create a certificate signing request (CSR)
- Create certificate from CSR with CA

- Configure web server to use certificate and TPM for private key operation
- Go for it!
Action
What you can do right now

- Do your applications speak TLS?
- If so, are they doing certificate validation?
- Where are the private keys stored and managed?
What you can do in the next months

- Consider your attacker is an insider
  - Which backend applications accept connections?
- Suppose there is a firewall or VPN misconfiguration
  - Is any data is exposed?
  - What authentication is your database using?
What you can do in the next months

- Once TLS is activated, make sure it is configured properly
  - Certificate validation
  - TLS 1.2

- Start using C or Go services built on open source tools