TLS Everywhere Made Practical

Or: making X.509 honour the Geneva Convention

SMARKETS

Mika Boström <mika.bostrom@smarkets.com>

VPE, Infrastructure Engineer, Appointed Information Security Officer™, Systems Wizard

Introduction - Smarkets

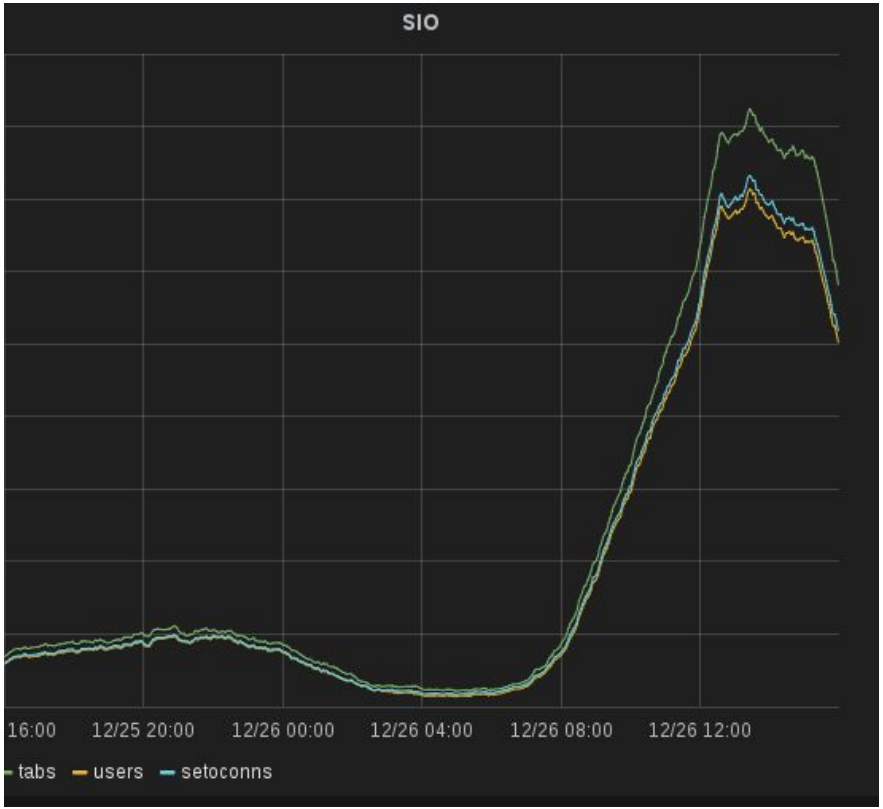
- Betting Exchange
- ~90 employees, >45 in engineering; 26 nationalities
- We handle other people's money
- Gambling is traditionally a shady industry with dubious actors
- Tech company with a devops culture: *you build it, you run it, you own it*
- Security must not be a premium account feature
- TLS Everywhere!

The Very Near Future

- Background for TLS everywhere
- Ups and downs
- Being your own CA
- Tooling
- Lessons learned

- Frequently Questioned Answers

Picture Quiz



Original Talk Idea...

Logging

Logging Is A Solved Problem

The Real Presentation

Why TLS?

- Data & Message Integrity
- Mutual Authentication (hello, client certs!)
- Increased Transport Security
- Prevents Trivial Traffic Dumping
- Regulator Demands

... on the other hand...


THAT HACK IS SO OLD
YOUR BIRTH CERTIFICATE SAYS "EXPIRED"
memegenerator.net

Unexpected Benefits Of Certificate Expiration

- Ugly Failure Mode
  - Long-term Monitoring = Essential
- `SSL_CERT_VERIFY_FAILED`
  - WTF?
  - Teaches to be verbose about obscure / unknown errors
- Encourages To Recycle Infrastructure
  - Persistent Services
  - Transient Worker Nodes

TLS Certificate Expiration:

- Makes Everyone Feel Relational DB Pain
  - Persistent DB connections are the norm
  - *“The DB Will Never Fail”* -- hubris of highest order
- REALLY Makes You Think Twice About Databases
  - (*eventually*)

TLS Everywhere

Let’s Build An Automatic Certificate Authority

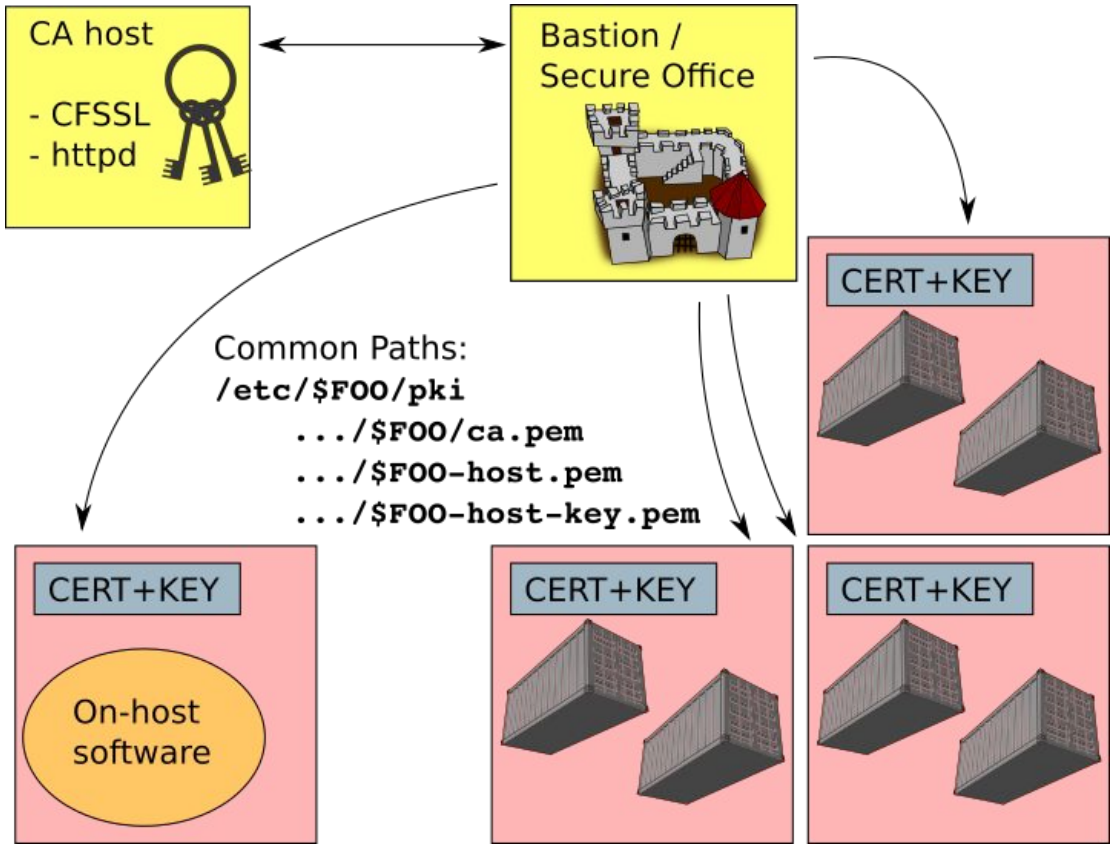
- CFSSL (https://github.com/cloudflare/cfssl)
- Build locally for CA system
  - Self contained - must not depend on CI
  - Building rest of infrastructure needs CA
- Binaries and root certificate easily available (non-CGI httpd rocks)
- Issuing host cert+key:

```
cfssl gencert -remote $CAHOST:$PORT
-hostname=[name1,name2,...] path/to/CSR.json | cfssljson
-bare $TARGETHOST
```

CSR.json

```
{
    "hosts": ["this.host.does.not.exist"], // overridden
    "key": { "algo": "rsa", "size": 2048 },
    "names": [
        {
            "C": "GB",
            "L": "London",
            "O": "Smarkets",
            "OU": "Smarkets Infrastructure"
        }
    ]
}
```

TLS Everywhere
Basic Architecture



Tooling: Abstracted In Ansible

```
roles:
    - { role: tls-certificate }
```

- Command line argument to **`ansible-playbook`** forces reissue
- Provisioning a certificate takes one line of configuration and one command
  - Applies to all hosts in target group
  - Idempotent, unless reissuing
- Configuring service for TLS takes longer than issuing and deploying certs
- Engineers’ client certificates almost as easy

The Bad & The Ugly

- ELB healthchecks with TLS
  - HTTP status -- **301 != 200**
  - Strict conformance = no HTTPS for your checks
- CA is always a global wildcard
  - CA cannot be valid only for: `*.internal.tld`
  - In other words: Name Constraints are not supported
    - *RFC 5280, §4.2.1.10 (`OID 2.5.29.30`) doesn't work*
- Tradeoff between very short lifetimes and maintaining an internal revocation list

Recap

- TLS does not need to be difficult
- Good tooling is crucial
- Expiration can help encourage better engineering practices
- **Especially** good for immutable infrastructure
- Bootstrapping requires a self-contained solution
- Recycling CA root certificate still an open question
- Bonus: CFSSL is superior for generating CSRs for public certs
    - OpenSSL CLI = usability nightmare

Final Slide

- Yes, we're hiring (who isn't?) - https://smarkets.com/careers
- Company Blog: https://smarketshq.com
- We don't interrupt user experience; we don't interrupt engineers' workflow
- Interested in learning more? Drop me a line

Mika Bostrom <mika.bostrom@smarkets.com>