TLS Vulnerability

- Utilized by large companies
  - paypal
  - amazon
  - chase bank
- susceptible to man-in-the-middle attacks

Why is TLS Vulnerable?

"the root cause of most of these vulnerabilities is the terrible design of the APIs to the underlying SSL libraries"

- libssl has 504 different symbols for developers to use!
- trying to securely connect to a site and download a few bytes securely took over 300 lines of code

Can we use the POSIX socket API to improve on this issue?

```
int socket = socket(PF_INET, SOCK_STREAM, IPPROTO_TLS)
```

- the third parameter is used to specify the protocol and Mark sought out to extend this to use the TLS protocol
- in the network subsytem, the kernel would actually make the call via TLS
- TLS under the hood actually calls TCP but in a secure way
- number of different functions was reduced from 504 to 14

What SSA Offers to developers

- setsockopt
- getsockopt

with TLS, you can make a socket and use the above functions to perform tasks needed for secure connections and retrieve certificates and data from other machines.

- administrator options
  - Global configuration file assigns TLS defaults
  - per-application profiles are available for custom settings
- TrustBase - an OS service that validates certificates according to admin config
- can enable multiple certificate checking services based on the user's needs

Reconfiguring From TLS to SSA

- took completely unfamiliar developers roughly 5 hrs to reconfigure sites already using openSSL to SSA!
- sites that were not currently using TCP took only roughly 10 minutes to set up!
- Mark's team made an openSSL emulator the would actually interface with SSA making it easy to have the system dynamically port a system over to SSA

Time Comparison

- no major time differences between TSL and SSA during stress test

Outcomes

- TLS through a known API
- admin control of TLS settings
- easy language support
- natural privilege separation
- alternative implementations supported