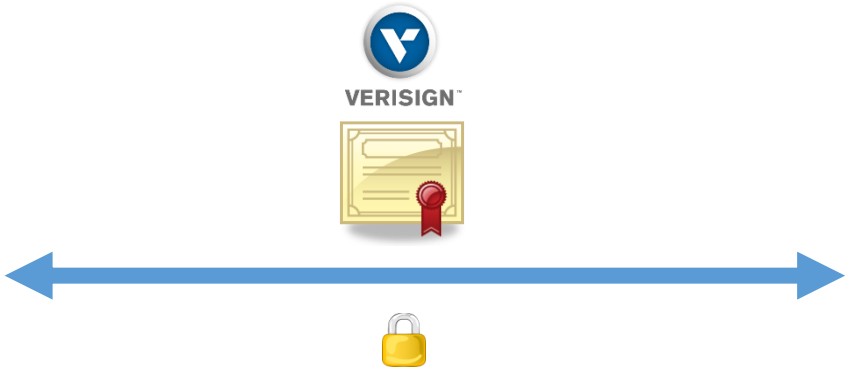
TrustBase

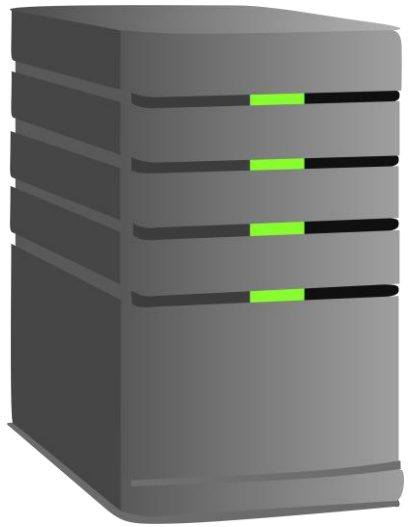
Repairing and Strengthening Certificate-based Authentication

Mark O'Neill, Scott Heidbrink, Scott Ruoti, Jordan Whitehead, Dan Bunker, Luke Dickinson, Travis Hendershot, Joshua Reynolds, Kent Seamons, and Daniel Zappala

client



amazon



certificate validation problems

certificate authorities (CAs)

- generally can sign certificates for any host (Eckersley et al.)
- have been hacked, sometimes repeatedly (Marlinspike)
- can be influenced and operated by governments (Soghoian et al.)
- don’t always follow best practices (see CNNIC)

for application developers

- mobile and desktop apps have validation problems
  - Brubaker et al., Georgiev et al., Onwuzurike et al., Fahl et al.
- security libraries are complicated
- security may not be a priority

`SSL_CTX_set_verify()`

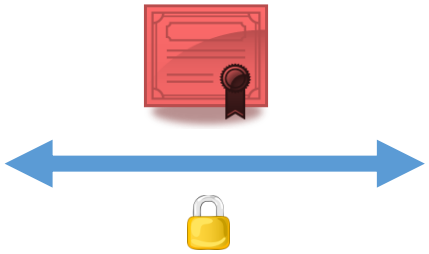
`X509_verify_cert()`

`SSL_CTX_set_cert_verify_callback()`

threat model

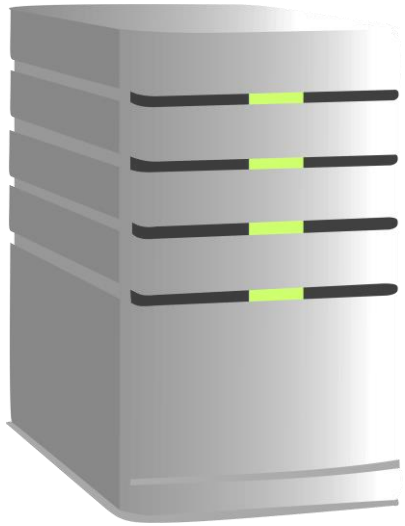
client

Bad guys, Inc.



amazon

VERISIGN



alternate and reinforcing strategies

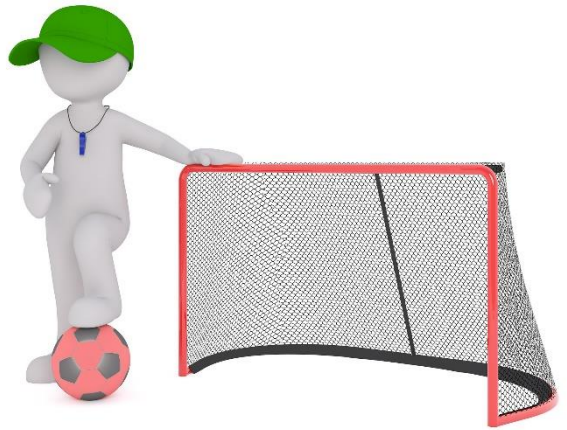
- mitigate many of these issues
- have no common platform or API
- have difficulty being adopted

trust decisions are outsourced

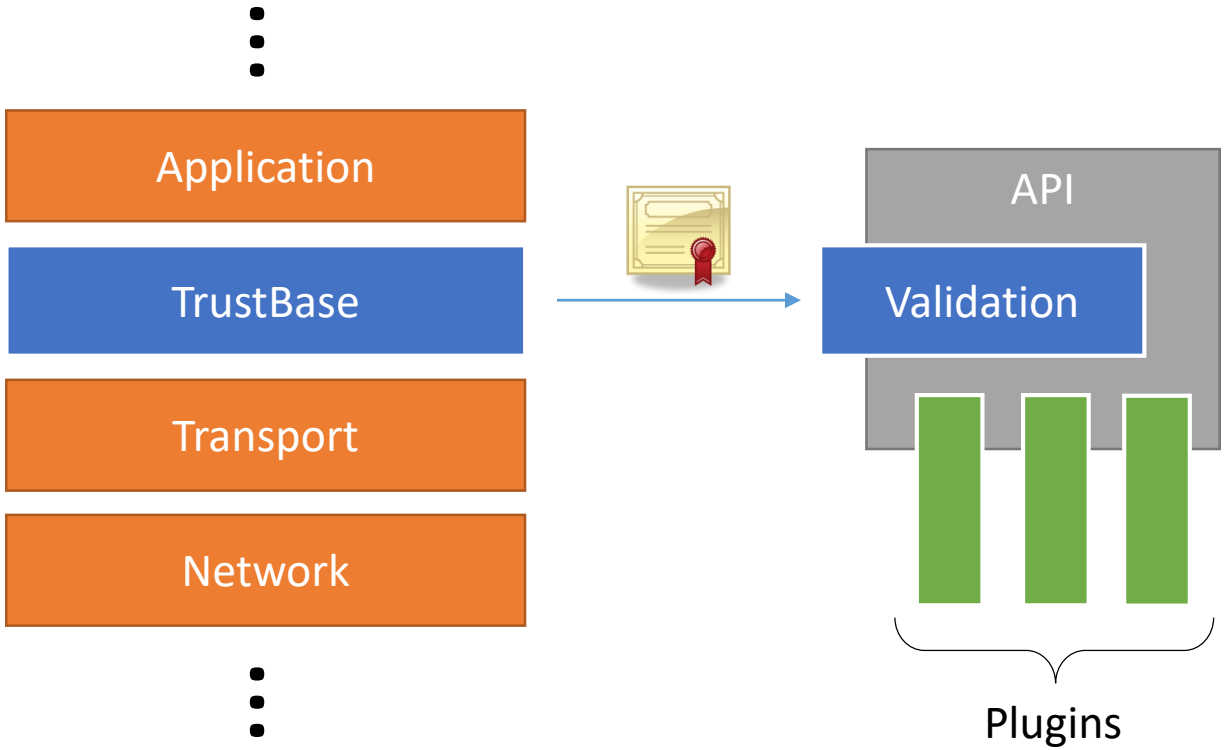
how do we enable admins to control trust decisions for their own machines?

TrustBase

- motivating principles
  - centralize authentication as an OS service
  - empower system admins to dictate how trust decisions are made
- design goals
  - secure *all* existing applications
  - prohibit unprivileged applications from acting against administrator rules
  - provide easy deployment of authentication systems
  - negligible overhead



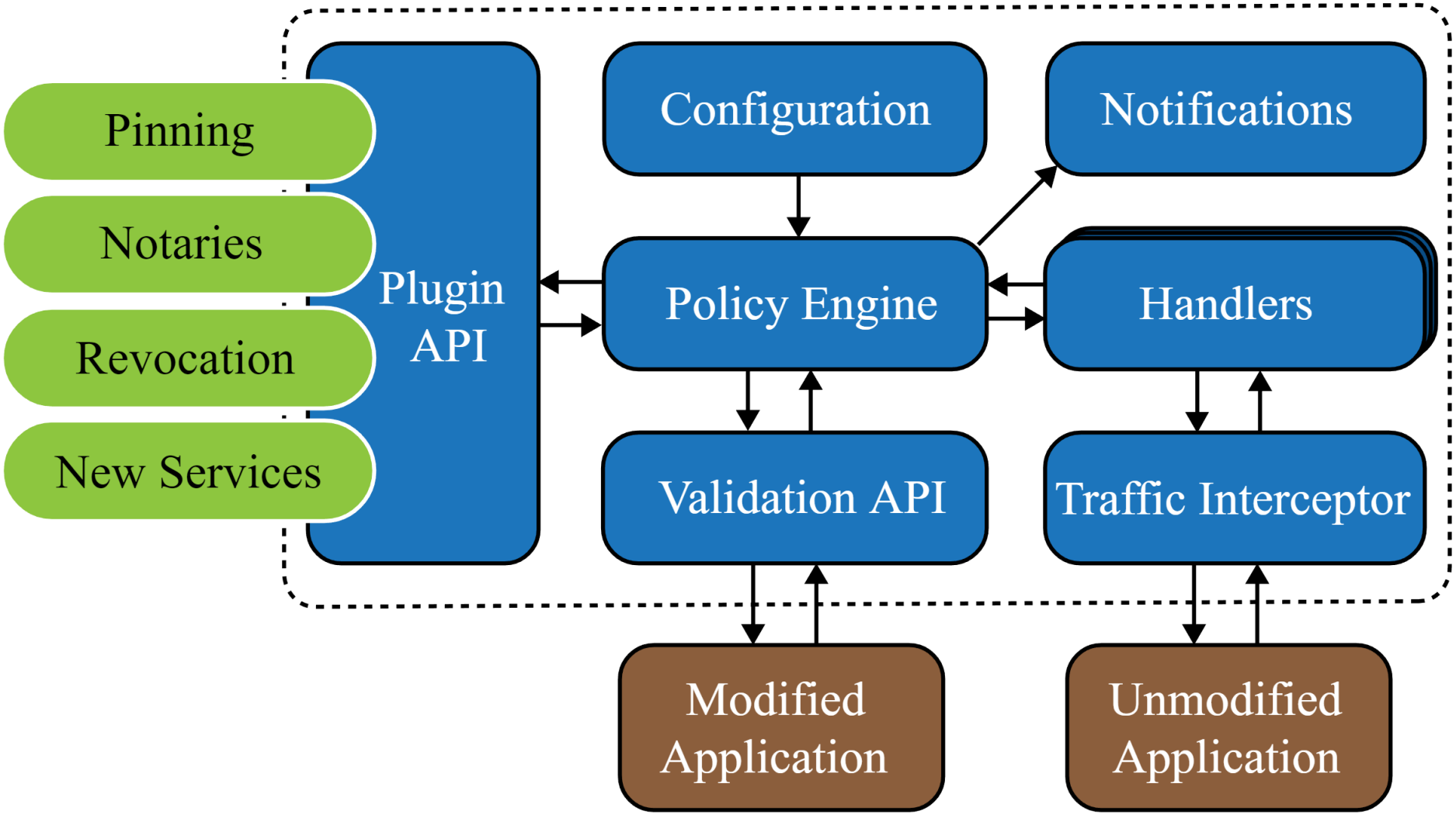
moving trust to the OS



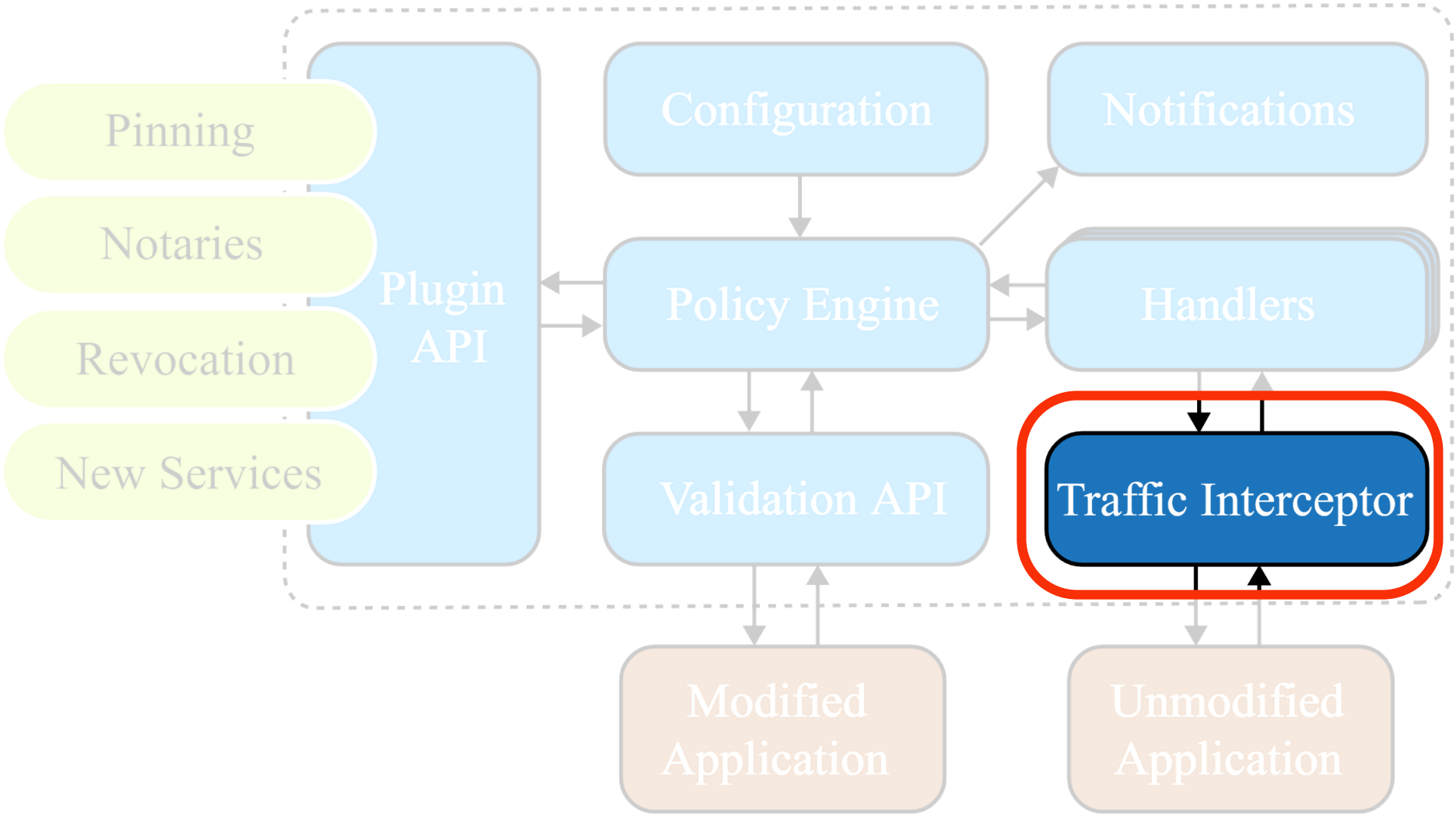
Prototypes for

- Linux
- Android (nonrooted)
- Windows

TrustBase architecture

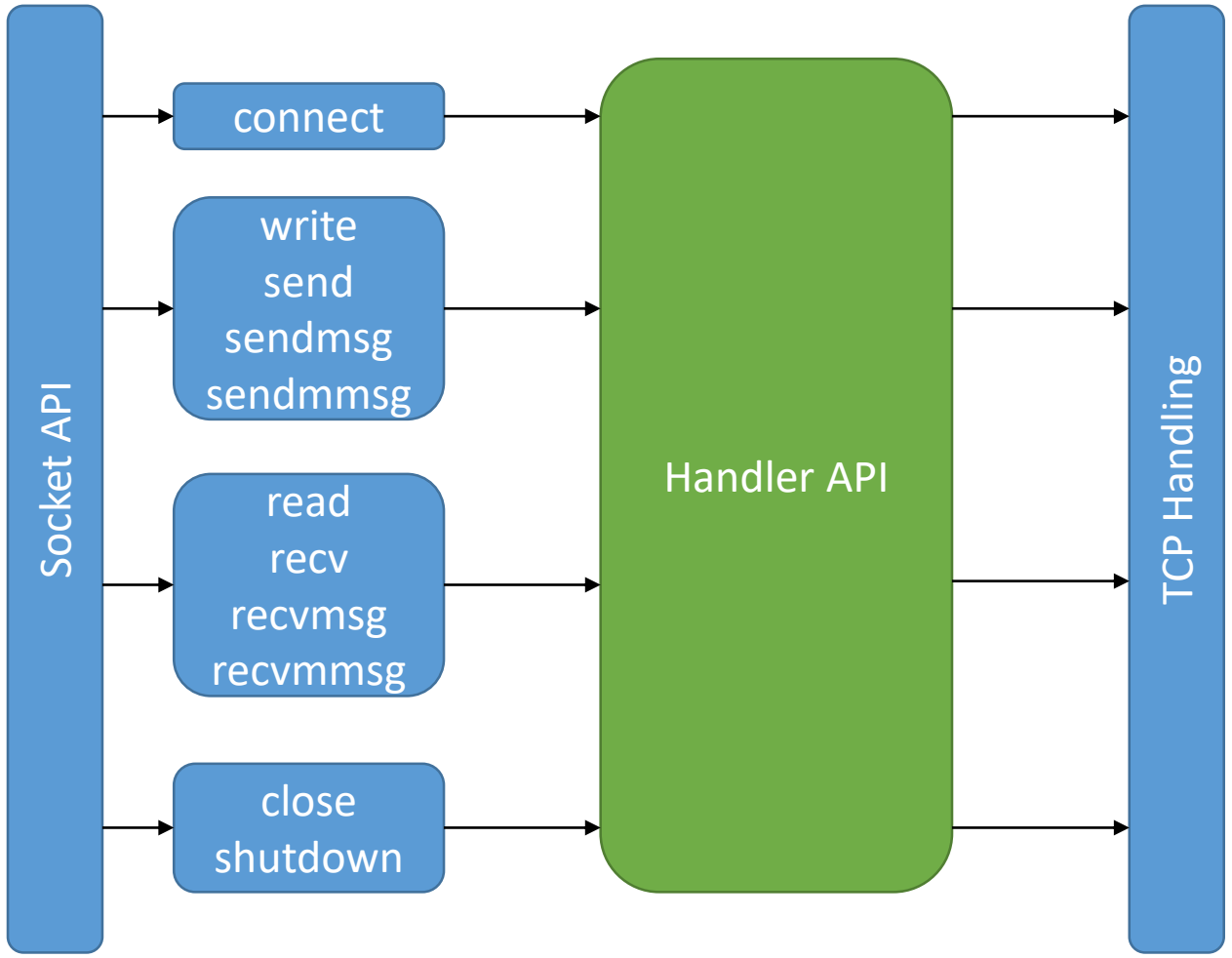


TrustBase architecture

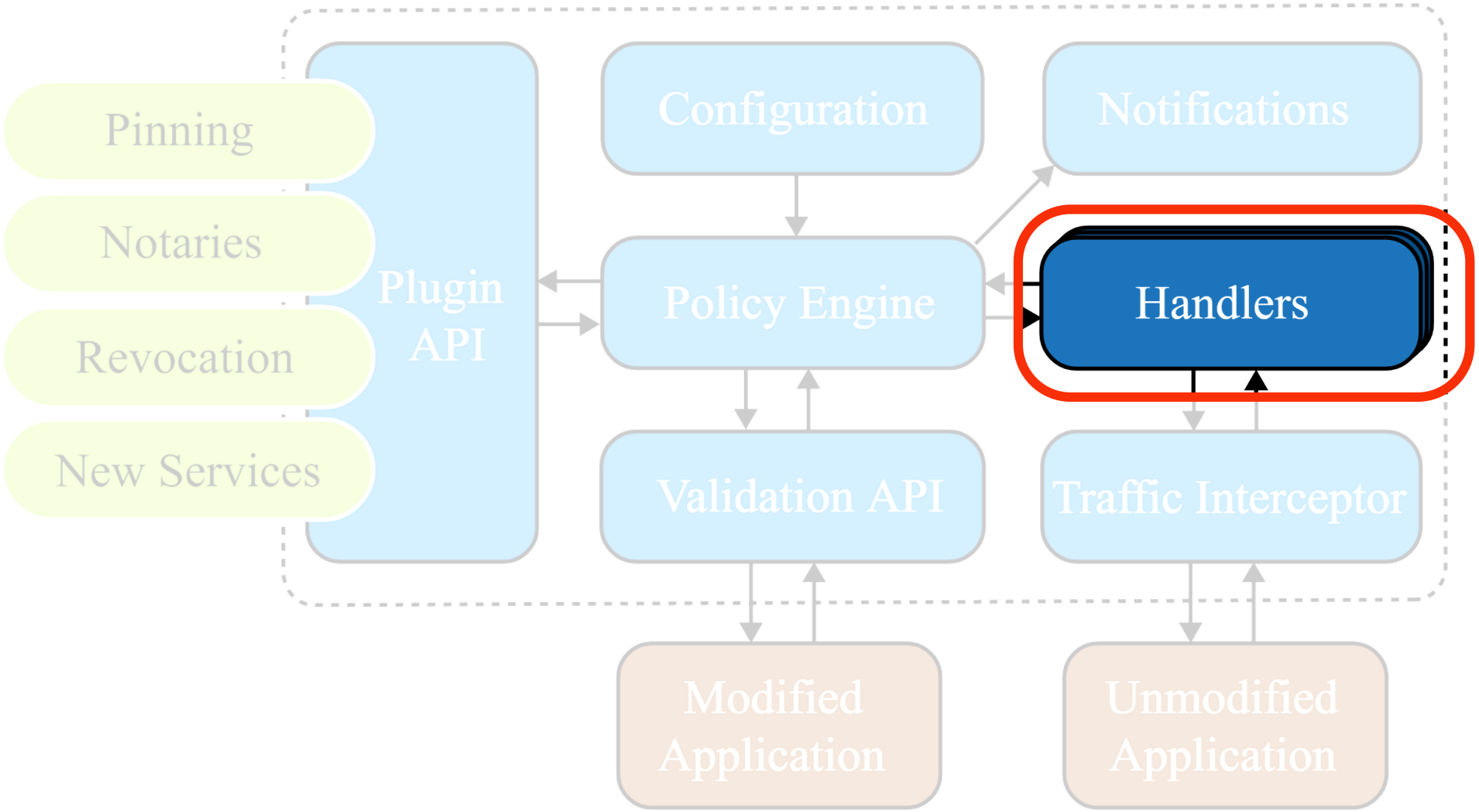


traffic interception (Linux)

- loadable kernel module
- hooks into native transport protocol functionality
- provides generic inspection/modification API



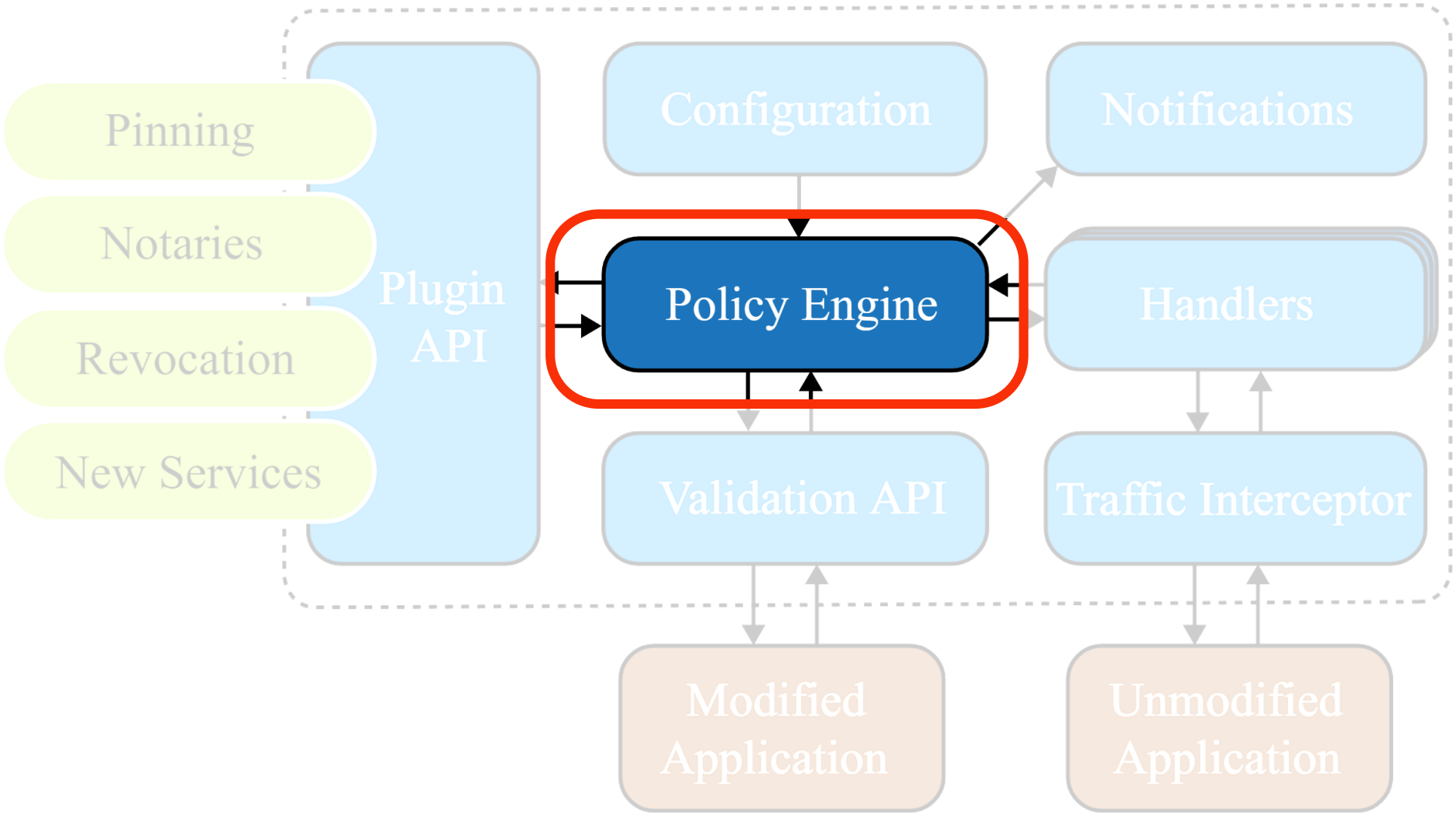
TrustBase architecture



TLS handler

1. monitor traffic for TLS records
2. record handshake messages
3. query policy engine with handshake data
4. receive policy response
    1. block connection if invalid
    2. allow if valid

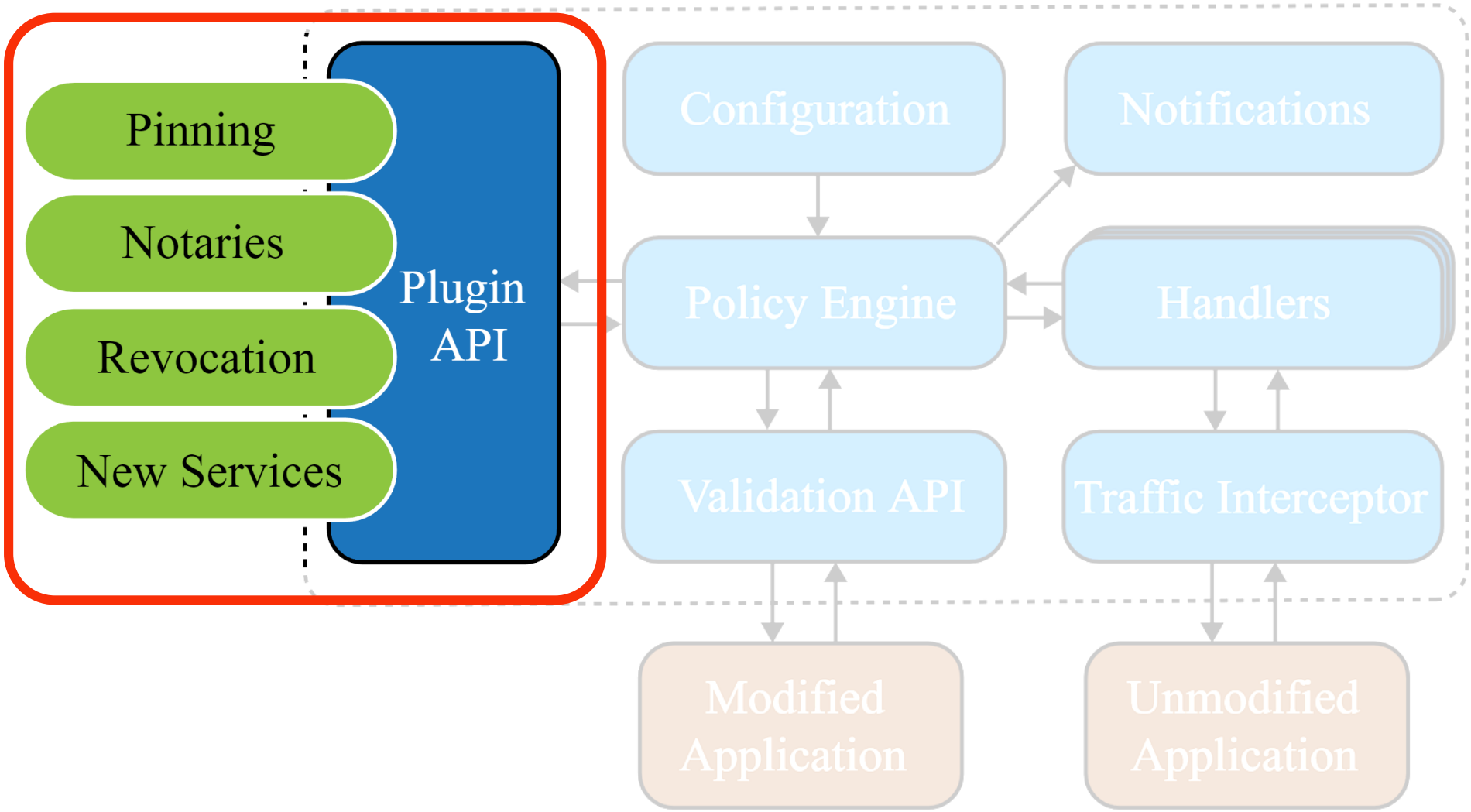
TrustBase architecture



policy engine

- receives queries via Netlink
- implements basic CA validation
- aggregates decisions from plugins
  - necessary
  - voting
- provides native API
  - Linux capabilities

TrustBase architecture

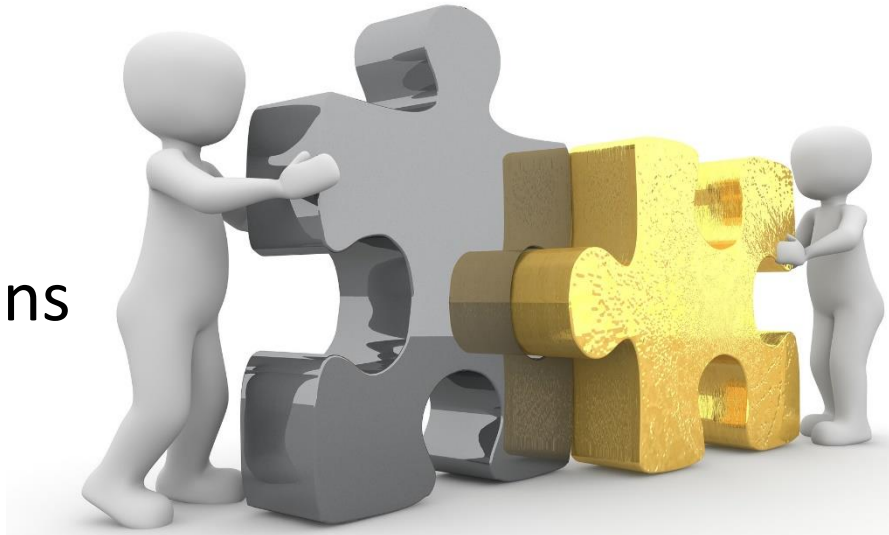


plugins

- API allows synchronous and asynchronous plugins
  - openssl STACK_OF(X509) or ASN.1 DER
- can report back yes/no/abstain/error for each chain
- have access to all handshake info (and more)

addons

- provide additional language support for plugins
- currently have native C and python addons
- API to add additional language support



included plugins and uses

- CA validation (builtin)
- certificate pinning
- OSCP checking
- CRLSet blocking
- DANE
- notary
- cipher suite auditor

evaluation

centralization and coverage

con

- single point of failure

pro

- updates are global
- benefit of many eyes

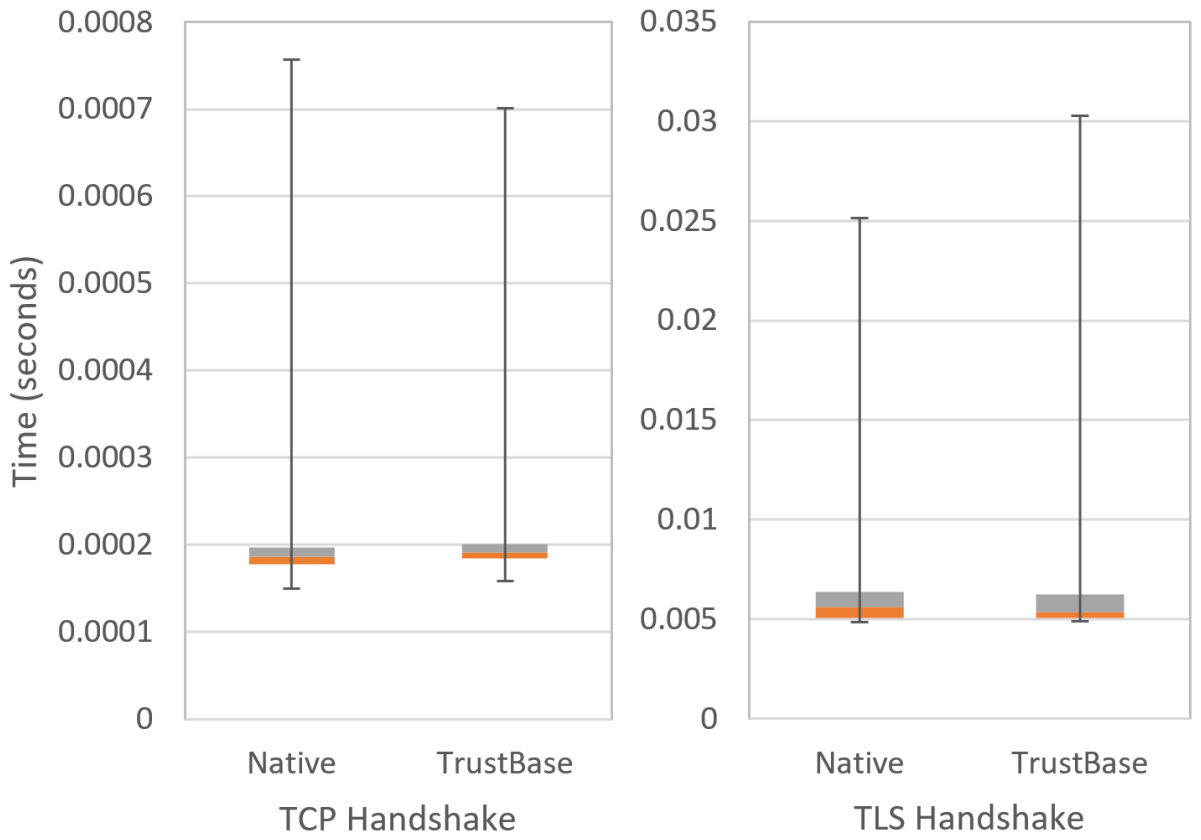
- TrustBase makes connection security an OS service, like TCP, IP

Library	Tool
C++	gnutls-cli
libcurl	curl
libgnutls	sslscan
libssl	openssl s_client
libnss	openssl s_time
JAVA	lynx
SSLSocketFactory	fetchmail
PERL	firefox
socket::ssl	chrome/chromium
PHP	mpop
fsockopen	w3m
php_curl	ncat
PYTHON	wget
httplib	steam
httplib2	thunderbird
pycurl	kmail
pyOpenSSL	pidgin
python ssl	
urllib, urllib2, urllib3	
requests	

hardening

- unprivileged malware cannot unload interception
- CAP_NET_RAW is required to use raw sockets (default) and to bypass TrustBase interception
- CAP_NET_ADMIN required to receive and respond to queries
- configuration is writable only by privileged users
- daemons run nonroot with only required permissions

performance



future work

- POSIX-based secure socket API
- push all of TLS to admin/OS control
- ease developer burden further
- call TrustBase validation natively
- wouldn’t this be nice?

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

trustbase lets **you** trust

who you want

how you want

Thank You

- source code: https://github.com/markoneill/trustbase-linux


Linux


Android


Win10

- pull requests welcome!
- project website: https://owntrust.org
- contact me: mto@byu.edu

- thanks to our sponsors:

Homeland Security

Sandia National Laboratories