Reconciling Performance and Security in High Load Environments

Ignat Korchagin
@ignatkn
$ whoami

- Performance and security at Cloudflare
- Passionate about security and crypto
- Enjoy low level programming
Performance vs Security
Performance vs Security
Performance vs Security

performance

security
Performance **AND** Security
Performance definition

• performance in the narrow sense
  ○ speed
  ○ throughput
  ○ latency
Performance definition

- performance in the narrow sense
  - speed
  - throughput
  - latency
- performance in the broader sense
  - all above
  - resource optimisation
  - process optimisation
  - etc
0-cost security
0-cost security

● security cost is negligible and/or affects some non-primary metric
0-cost security

- security cost is negligible and/or affects some non-primary metric
- security cost is hidden/amortised by the architecture or implementation
0-cost security

- security cost is negligible and/or affects some non-primary metric
- security cost is hidden/amortised by the architecture or implementation
- the cost is not incurred for normal system behaviour (prohibitive security)
Negligible security cost: secure boot chain

system firmware
Negligible security cost: secure boot chain

```
  system firmware
    ▼ verify
     bootloader
```
Negligible security cost: secure boot chain

- system firmware
  - verify
- bootloader
  - verify
- operating system
Negligible security cost: secure boot chain
Negligible security cost: secure boot chain

- system firmware
- bootloader
- operating system
- drivers
- applications, services
Negligible security cost: secure boot chain

- System firmware
- Bootloader
- Operating system
- Drivers
- Applications, services
Negligible security cost: secure boot chain

![Diagram showing the secure boot chain with verified system firmware, bootloader, operating system, drivers, and applications/services.](image-url)
Negligible security cost: secure boot chain

● ensures all running code is authorised by the system owner
Negligible security cost: secure boot chain

• ensures all running code is authorised by the system owner
• most effective protection from persistent malware
Negligible security cost: secure boot chain

- ensures all running code is authorised by the system owner
- most effective protection from persistent malware
- enforces operational procedures
  - all changes are properly fixed in the VCS
  - no possibility for one-off fixes
  - systems run only what’s needed
Negligible security cost: secure boot chain

- ensures all running code is authorised by the system owner
- most effective protection from persistent malware
- enforces operational procedures
  - all changes are properly fixed in the VCS
  - no possibility for one-off fixes
  - systems run only what’s needed
- affects system boot time only
  - adds at most ~ms boot time
Amortised security cost: data encryption at rest
Amortised security cost: data encryption at rest

- applications
- filesystems
Amortised security cost: data encryption at rest

- applications
- filesystems
- block subsystem
Amortised security cost: data encryption at rest

- applications
- filesystems
- block subsystem
- storage hardware
Amortised security cost: data encryption at rest

- Applications
- Filesystems
- Block subsystem
- Storage hardware

SED, OPAL
Amortised security cost: data encryption at rest

- **applications**
- **filesystems**
- **block subsystem**
- **storage hardware**

LUKS/dm-crypt, BitLocker, FileVault

SED, OPAL
Amortised security cost: data encryption at rest

applications

filesystems

block subsystem

storage hardware

LUKS/dm-crypt, BitLocker, FileVault

dcryptfs, ext4 encryption or fscrypt

SED, OPAL

@ignatkn
Amortised security cost: data encryption at rest

- **Applications**: DBMS, PGP, OpenSSL, Themis
- **Filesystems**: LUKS/dm-crypt, BitLocker, FileVault
- **Block subsystem**: encryptfs, ext4 encryption or fscrypt
- **Storage hardware**: SED, OPAL

@ignatkn
Amortised security cost: data encryption at rest

- applications
  - DBMS, PGP, OpenSSL, Themis
- filesystems
  - LUKS/dm-crypt, BitLocker, FileVault
  - encryptfs, ext4 encryption or fscrypt
- block subsystem
- storage hardware
  - SED, OPAL

@ignatkn
Amortised security cost: data encryption at rest

Advantages of OS full disk encryption
Amortised security cost: data encryption at rest

Advantages of OS full disk encryption

- little configuration needed
Amortised security cost: data encryption at rest

Advantages of OS full disk encryption

- little configuration needed
- fully transparent to applications
Amortised security cost: data encryption at rest

Advantages of OS full disk encryption

- little configuration needed
- fully transparent to applications
- don’t roll our own crypto (unlike application layer)
Amortised security cost: data encryption at rest

Advantages of OS full disk encryption

- little configuration needed
- fully transparent to applications
- don’t roll our own crypto (unlike application layer)
- open, audible (unlike hardware layer)
What is a CDN?

https://en.wikipedia.org/wiki/Content_delivery_network
What is a CDN?

https://en.wikipedia.org/wiki/Content_delivery_network
Cloudflare Network
Average CDN cache response tail latency

- encrypted (vanilla LUKS/dm-crypt)
- unencrypted

@ignatkn
Average CDN cache response tail latency

- encrypted (vanilla LUKS/dm-crypt)
- unencrypted
- encrypted (patched LUKS/dm-crypt)

https://www.usenix.org/conference/vault20/presentation/korchagin
Disk encryption overhead

- expected lower disk encryption overhead
  - got none
  - no changes in crypto algorithms, formats etc
Disk encryption overhead

● expected lower disk encryption overhead
  ○ got none
  ○ no changes in crypto algorithms, formats etc
● zero overhead data encryption is a no-brainer
Disk encryption overhead

- expected lower disk encryption overhead
  - got none
  - no changes in crypto algorithms, formats etc
- zero overhead data encryption is a no-brainer
- encourages further performance improvement research
  - data encryption is not the bottleneck anymore
  - indicates potential room for performance improvements for the overall system
Prohibitive security: syscalls

application
Prohibitive security: syscalls

application

OS kernel
Prohibitive security: syscalls

application

OS kernel

open  read  write  send  recv  accept
Prohibitive security: syscalls

application

OS kernel

open  read  write  send  recv  accept
Prohibitive security: syscalls and seccomp

- open
- read
- write
- send
- recv
- accept

Contract:
- open
- read
- write
Prohibitive security: syscalls and seccomp

- open
- read
- write
- send
- recv
- accept

Contract
- open
- read
- write

@ignatkn
Prohibitive security: syscalls and seccomp

Application

Open
Read
Write
Send
Recv
Accept

OS kernel

Contract
- Open
- Read
- Write

@ignatkn
Prohibitive security: syscalls and seccomp
Prohibitive security: syscalls and seccomp
Prohibitive security: syscalls and seccomp

Hi! I’m a clock app. I will only use `gettimeofday`
Prohibitive security: syscalls and seccomp

Hi! I’m a clock app. I will only use `gettimeofday`
Prohibitive security: syscalls and seccomp

Hi! I’m a clock app. I will only use `gettimeofday`

`gettimeofday`

1970-01-01T00:00:00Z
Prohibitive security: syscalls and seccomp

Hi! I’m a clock app. I will only use `gettimeofday`

`gettimeofday`

1970-01-01T00:00:00Z
Prohibitive security: syscalls and seccomp

Hi! I’m a clock app. I will only use `gettimeofday`
Prohibitive security: syscalls and seccomp

Hi! I’m a clock app. I will only use gettimeofday

gettimeofday
1970-01-01T00:00:00Z

send

@ignatkn
Prohibitive security: syscalls and seccomp

• greatly limits the potential damage of RCE exploits
Prohibitive security: syscalls and seccomp

- greatly limits the potential damage of RCE exploits
- 0-cost overhead
  - no security is triggered for expected system behaviour
Prohibitive security: syscalls and seccomp

- greatly limits the potential damage of RCE exploits
- 0-cost overhead
  - no security is triggered for expected system behaviour
- improves development velocity
  - developer intent vs actual implementation
Security and systems performance
HTTP/2 and HTTP/3

HTTP/2 (2015)
HTTP/2 and HTTP/3

HTTP/2 (2015)
- major rework from HTTP/1 (1991)
  - binary protocol
  - connection multiplexing
  - server push
HTTP/2 and HTTP/3

HTTP/2 (2015)
- major rework from HTTP/1 (1991)
  - binary protocol
  - connection multiplexing
  - server push

HTTP/3 (in progress)
- transport over QUIC/UDP
HTTP/2 performance (2015)

<table>
<thead>
<tr>
<th>Access via HTTP Protocol Version</th>
<th>Average Page Load time</th>
</tr>
</thead>
<tbody>
<tr>
<td>HTTP 1.x</td>
<td>9.07 sec.</td>
</tr>
<tr>
<td>SPDY/3.1</td>
<td>7.06 sec.</td>
</tr>
<tr>
<td>HTTP/2</td>
<td>4.27 sec.</td>
</tr>
</tbody>
</table>

HTTP/2 performance

https://imagekit.io/demo/http2-vs-http1
HTTP/2 performance

https://www.flickr.com/photos/smemon/15944989872/
HTTP/2 performance

https://www.flickr.com/photos/smemon/15944989872/

@ignatkn
SSL/TLS

SSL/TLS: RSA vs ECC


@ignatkn
SSL/TLS: RSA vs ECC

- RSA
  - “older” cryptosystem (1977)
  - factoring problem of large numbers
  - sub-exponential complexity cracking algorithms
  - large keys (>=2048 bit)
SSL/TLS: RSA vs ECC

● RSA
  ○ “older” cryptosystem (1977)
  ○ factoring problem of large numbers
  ○ sub-exponential complexity cracking algorithms
  ○ large keys (>=2048 bit)

● ECC
  ○ “newer” cryptosystem (1985)
  ○ discrete logarithm problem over elliptic curves
  ○ exponential complexity cracking algorithms
  ○ small keys (>=256 bit)
SSL/TLS: RSA vs ECC

$ openssl speed rsa ecdsa
SSL/TLS: RSA vs ECC

<table>
<thead>
<tr>
<th>Cipher</th>
<th>Sign Time (s)</th>
<th>Verify Time (s)</th>
<th>Sign Rate (Ops/s)</th>
<th>Verify Rate (Ops/s)</th>
</tr>
</thead>
<tbody>
<tr>
<td>rsa 2048 bits</td>
<td>0.000616s</td>
<td>0.000018s</td>
<td>1623.5</td>
<td>55200.6</td>
</tr>
<tr>
<td>256 bit ecdsa (nistp256)</td>
<td>0.0000s</td>
<td>0.0001s</td>
<td>25487.6</td>
<td>10731.6</td>
</tr>
</tbody>
</table>
SSL/TLS: RSA vs ECC

$ openssl speed rsa ecdsa

<table>
<thead>
<tr>
<th></th>
<th>sign</th>
<th>verify</th>
<th>sign/s</th>
<th>verify/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>rsa 2048 bits</td>
<td>0.000616s</td>
<td>0.000018s</td>
<td>1623.5</td>
<td>55200.6</td>
</tr>
<tr>
<td>256 bit ecdsa (nistp256)</td>
<td>0.0000s</td>
<td>0.0001s</td>
<td>25487.6</td>
<td>10731.6</td>
</tr>
</tbody>
</table>

- faster TLS handshakes (~15 times faster from above)
SSL/TLS: RSA vs ECC

$ openssl speed rsa ecdsa

<table>
<thead>
<tr>
<th></th>
<th>sign (s)</th>
<th>verify (s)</th>
<th>sign/s</th>
<th>verify/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>rsa 2048 bits</td>
<td>0.000616</td>
<td>0.000018</td>
<td>1623.5</td>
<td>55200.6</td>
</tr>
<tr>
<td>256 bit ecdsa (nistp256)</td>
<td>0.0000</td>
<td>0.0001</td>
<td>25487.6</td>
<td>10731.6</td>
</tr>
</tbody>
</table>

- faster TLS handshakes (~15 times faster from above)
- less CPU utilisation
## SSL/TLS: RSA vs ECC

<table>
<thead>
<tr>
<th>Command</th>
<th>sign/s</th>
<th>verify/s</th>
<th>sign/s</th>
<th>verify/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>openssl speed rsa ecdsa</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>rsa 2048 bits</td>
<td>0.000616s</td>
<td>0.000018s</td>
<td>1623.5</td>
<td>55200.6</td>
</tr>
<tr>
<td>256 bit ecdsa (nistp256)</td>
<td>0.0000s</td>
<td>0.0001s</td>
<td>25487.6</td>
<td>10731.6</td>
</tr>
</tbody>
</table>

- faster TLS handshakes (~15 times faster from above)
- less CPU utilisation
- less key storage
SSL/TLS: RSA vs ECC

$ openssl speed rsa ecdsa

<table>
<thead>
<tr>
<th></th>
<th>sign/s</th>
<th>verify/s</th>
</tr>
</thead>
<tbody>
<tr>
<td>rsa 2048 bits</td>
<td>0.000616s</td>
<td>0.000018s</td>
</tr>
<tr>
<td>256 bit ecdsa (nistp256)</td>
<td>0.00000s</td>
<td>0.0001s</td>
</tr>
</tbody>
</table>

- faster TLS handshakes (~15 times faster from above)
- less CPU utilisation
- less key storage
- better security
SSL/TLS: RSA vs ECC (2017)

Negotiated TLS key exchange algorithms:
- ECDHE (x25519), 30.5%
- ECDHE (P256), 68.3%
- RSA, 0.6%

Negotiated TLS signature algorithms:
- RSA, 24.9%
- ECDSA (P256), 75.1%

% of BoringSSL CPU time:
- P256, 8%
- x25519, 4%
- RSA, 49%

https://blog.cloudflare.com/how-expensive-is-crypto-anyway/
The Internet: network of networks

The Internet: AS and BGP

I have 1.1.1.1
The Internet: AS and BGP

I have 1.1.1.1

I have 8.8.8.8
The Internet: packet switching

The original message is Green, Blue, Red.
The Internet: BGP security

I have 1.1.1.1

I have 8.8.8.8
The Internet: BGP security

I have 1.1.1.1

I have 8.8.8.8

I have 1.2.3.4
The Internet: BGP security

I have 1.1.1.1
I have 1.2.3.4

I have 8.8.8.8
I have 1.2.3.4
The Internet: BGP with RPKI

I have 1.1.1.1

I have 1.2.3.4

I have 8.8.8.8

I have 1.2.3.4

@ignatkn
The Internet: BGP with RPKI

I have 1.1.1.1

I have 8.8.8.8

I have 1.2.3.4

I have 1.2.3.4

@ignatkn
The Internet: BGP with RPKI

- RPKI prevents bad actors from claiming resources they don’t own
The Internet: BGP with RPKI

- RPKI prevents bad actors from claiming resources they don’t own
- however, not all “false claimers” are bad actors
  - bugs in network equipment software
  - network equipment misconfigurations
The Internet: BGP with RPKI

- RPKI prevents bad actors from claiming resources they don’t own
- however, not all “false claimers” are bad actors
  - bugs in network equipment software
  - network equipment misconfigurations
- RPKI improves network throughput by ensuring routes validity
  - some misconfigurations cause severe outages
  - minor misconfigurations create packet loss

Security and process performance
Datacentre provisioning

- connect hardware
Datacentre provisioning

● connect hardware
● verify hardware
Datacentre provisioning

- connect hardware
- verify hardware
  - setup initial network
Datacentre provisioning

- connect hardware
- verify hardware
  - setup initial network
  - configure OOB
Datacentre provisioning

- connect hardware
- verify hardware
  - setup initial network
  - configure OOB
  - secure OOB
Datacentre provisioning

- connect hardware
- verify hardware
  - setup initial network
  - configure OOB
  - secure OOB
  - dump serial numbers
Datacentre provisioning

- connect hardware
- verify hardware
  - setup initial network
  - configure OOB
  - secure OOB
  - dump serial numbers
  - cross-check with the inventory system
Datacentre provisioning

- connect hardware
- verify hardware
  - setup initial network
  - configure OOB
  - secure OOB
  - dump serial numbers
  - cross-check with the inventory system
- initial key provisioning
  - ssh and/or configuration management
  - verify and authorise key fingerprints
Datacentre provisioning

● connect hardware
● verify hardware
  ○ setup initial network
  ○ configure OOB
  ○ secure OOB
  ○ dump serial numbers
  ○ cross-check with the inventory system
● initial key provisioning
  ○ ssh and/or configuration management
  ○ verify and authorise key fingerprints
Datacentre provisioning

- connect hardware
- verify hardware
  - setup initial network
  - configure OOB
  - secure OOB
  - dump serial numbers
  - cross-check with the inventory system
- initial key provisioning
  - ssh and/or configuration management
  - verify and authorise key fingerprints

@ignatkn
What is a TPM?
What is a TPM?

- tamper resistant crypto chip in modern laptops and servers
What is a TPM?

- tamper resistant crypto chip in modern laptops and servers
- can provide secure key storage and hardware random number generator
What is a TPM?

- tamper resistant crypto chip in modern laptops and servers
- can provide secure key storage and hardware random number generator
- fundamental building block for remote attestation
  - authenticated identity for remote systems
  - trustworthy assertions about the state of the remote systems
Remote attestation

verifier
Remote attestation

verifier

remote host

TPM

@ignatkn
Remote attestation

verifier

quote

remote host

TPM
Remote attestation
Remote attestation

verifier

remote host

TPM

quote
Remote attestation

verify

quote

remote host

TPM
Remote attestation

- we’re communicating with the right host
Remote attestation

- we’re communicating with the right host
- we’re communicating with the right host securely
Remote attestation

- we’re communicating with the right host
- we’re communicating with the right host securely
- the remote host runs only authorised software
  - firmware
  - operating system
  - other software
Datacentre provisioning with TPM
Datacentre provisioning with TPM

- verify server identity
Datacentre provisioning with TPM

- verify server identity
- verify running OS
Datacentre provisioning with TPM

- verify server identity
- verify running OS
- cross-check serial numbers
Datacentre provisioning with TPM

- verify server identity
- verify running OS
- cross-check serial numbers
- provision configuration
- management keys
Datacentre provisioning with TPM

- verify server identity
- verify running OS
- cross-check serial numbers
- provision configuration management keys
- start serving production traffic
Datacentre provisioning with TPM

- verify server identity
- verify running OS
- cross-check serial numbers
- provision configuration management keys
- start serving production traffic
Datacentre provisioning with TPM

- verify server identity
- verify running OS
- cross-check serial numbers
- provision configuration management keys
- start serving production traffic
Datacentre provisioning with TPM

- better automation
  - less room for human errors or misconfigurations
Datacentre provisioning with TPM

- better automation
  - less room for human errors or misconfigurations
- faster datacentre provisioning
  - from weeks to days
Datacentre provisioning with TPM

- **better automation**
  - less room for human errors or misconfigurations
- **faster datacentre provisioning**
  - from weeks to days
- **efficient engineering time**
  - engineers can develop/improve systems rather than do repetitive tasks
Datacentre provisioning with TPM

- **better automation**
  - less room for human errors or misconfigurations
- **faster datacentre provisioning**
  - from weeks to days
- **efficient engineering time**
  - engineers can develop/improve systems rather than do repetitive tasks
- **better security**
Conclusions

● security does not always have to impact performance
  ○ 0-cost security
● sometimes security actually improves performance
● security can improve performance in the broader sense and in longer term
● “performance by security” approach is useful in driving and prioritising company security improvements
Questions?