Security

Emerging technologies & open source

Mike Bursell
Chief Security Architect

Nathaniel McCallum
Senior Principal Software Engineer

May 2019

LEGAL DISCLAIMER

The content set forth herein does not constitute in any way a binding or legal agreement or impose any legal obligation or duty on Red Hat. This information is provided for discussion purposes only and is subject to change for any or no reason.

Intros

Mike Bursell

- Security geek
- Red Hatter
- Theologian

Nathaniel McCallum

- Security geek
- Red Hatter
- Theologian

Intros

Mike Bursell

- Security geek
- Red Hatter
- Theologian (Emacs, Protestantism)

Nathaniel McCallum

- Security geek
- Red Hatter
- Theologian (vi, Catholicism)

Security - or risk?

Security - or risk?

- Security isn’t an end in itself
- Security helps with risk
  - Monitoring, mitigating, measuring, managing

Security - or risk?

- Security isn’t an end in itself
- Security helps with risk
  - Monitoring, mitigating, measuring, managing
- Up and down the stack
  - From kernel…to UI
  - From developer to C-level

Security - or risk?

- Security isn't an end in itself
- Security helps with risk
  - Monitoring, mitigating, measuring, managing
- Up and down the stack
  - From kernel…to UI
  - From developer to C-level
- Through the lifecycle
  - Dev… through Ops… through auditing … through design…

Security - or risk?

Open source - what's different?

- Everybody can see everything
  - Both good and bad
- "Build or buy" is an option
- Risk is difficult to quantify
  - This is something vendors & community need to work on
  - Trade-offs - business vs security
  - Stack, infrastructure, component, even package-level
  - Cyber-insurance

Tech, process and culture

Tech
Basic

Tech is easy*

**yeah, I know: stick with me here*

Tech
Basic

Tech is easy

- We know how to implement technology
- Open source has won*

- Of course: security tech *isn't* easy
- But there are options, certifications and standards to help
  - (e.g. the excellent NIST-800 series)

*https://www.redhat.com/en/blog/survey-says-enterprise-open-source-inventing-future-software

Process

D&D Expert Edition

Process is harder

- Methodologies (DevSecOps, Kanban, Agile, Prince II)
- Project management (stand-ups, videocalls, IRC)
- Management structures (matrix, project-based, specialism)
- External-internal information split (open source, innersource)
- Lifecycle (CI/CD, automation, boundaries/gate checks…)

Culture

Who's *the Daddy? (AD&D Dungeon Master)*

Cultural changes are hard.

- Security folks…
  - Are "special"
  - Are known for "no"
  - Like their ivory towers

Culture

Who's *the Daddy? (AD&D Dungeon Master)*

Cultural changes are hard.

- Security folks…
  - Are "special"
  - Are known for "no"
  - Like their ivory towers
- But…
  - Need to be integrated
  - Don't scale
  - Require management support
  - Are real people, too

Culture

Who's *the Daddy? (AD&D Dungeon Master)*

Cultural changes are hard.

- Security folks…
  - Are "special"
  - Are known for "no"
  - Like their ivory towers
- But…
  - Need to be integrated
  - Don't scale
  - Require management support
  - Are real people, too

Shh! It’s a secret

Confidentiality and secrets

Keeping data secret

- Quantum computing
- Crypto-agility

And sharing it (selectively)

- Zero-sum proofs
- MPC (Multi-party computation)
- Homomorphic encryption

Confidentiality and secrets

Keeping data secret

- Quantum computing
- Crypto-agility

And sharing it (selectively)

- Zero-sum proofs
- MPC (Multi-party computation)
- Homomorphic encryption

...and open source is *vital.*

If we get left behind in standards, it will be a huge step backwards.

“All the world is made of faith, and trust,* and pixie dust”

“All the world is made of faith, and trust,* and pixie dust”

*(with apologies for the Oxford comma)

Trust

Vital for Open Hybrid Cloud

Trust definition

"Trust is the assurance that

one entity holds that

another will perform

particular actions

according to a specific expectation."

http://aliceevebob.com/2017/05/09/what-is-trust-with-apologies-to-pontius-pilate/

Trust

Vital for Open Hybrid Cloud

The fallacy of “zero-trust”

- E.g. blockchain

Trust
Vital for Open Hybrid Cloud

The fallacy of “zero-trust”

- E.g. blockchain
  - “What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party.”
  - Satoshi Nakamoto: bitcoin white paper (https://bitcoin.org/bitcoin.pdf).

Trust
Vital for Open Hybrid Cloud

The fallacy of “zero-trust”

- E.g. blockchain
  - “What is needed is an electronic payment system based on cryptographic proof instead of trust, allowing any two willing parties to transact directly with each other without the need for a trusted third party.”
  - Satoshi Nakamoto: bitcoin white paper (https://bitcoin.org/bitcoin.pdf).
  - Maybe not a run-time third party, but unless you audit and compile the code yourself (and the toolchain, and the infrastructure), then you have trust in a third party.

Move to “explicit trust”

Trust

Vital for Open Hybrid Cloud

Move to “explicit trust”

Trust

Vital for Open Hybrid Cloud

Move to “explicit trust”

"Trust is the assurance that

one entity holds that

another will perform

particular actions

according to a specific expectation."

Trust

Vital for Open Hybrid Cloud

Move to “explicit trust”

"Trust is the assurance that

one entity holds that

another will perform

particular actions

according to a specific expectation."

The Perimeter is dead.

Trust

Vital for Open Hybrid Cloud

Move to “explicit trust”

Explicit trust allows for risk calculations

Trust

Vital for Open Hybrid Cloud

Move to “explicit trust”

Explicit trust allows for risk calculations

Some of the pieces

- Policy
- Host attestation
- Hardware

The Perimeter is dead

What to trust, what not to trust?

Policy management

- Policies
  - Networking, authentication, crypto, workload scheduling, …
  - Define: Apply: Validate: Enforce

The Perimeter is dead

What to trust, what not to trust?

Policy management

- Policies
  - Networking, authentication, crypto, workload scheduling, …
  - Define: Apply: Validate: Enforce
  - “D.A.V.E.”

The Perimeter is dead

What to trust, what not to trust?

Policy management

- Policies
  - Networking, authentication, crypto, workload scheduling, …
  - Define: Apply: Validate: Enforce
  - “D.A.V.E.”
- Domain specific
- Definition language required
- Federation, hierarchies complex - danger of “firewalls problem”

The Perimeter is dead

What to trust, what not to trust?

Policy management

- Policies
  - Networking, authentication, crypto, workload scheduling, …
  - Define: Apply: Validate: Enforce
  - “D.A.V.E.”
- Domain specific
- Definition language required
- Federation, hierarchies complex - danger of “firewalls problem”

Example: Open Policy Agent (OPA)

https://www.openpolicyagent.org/

Open Policy Agent

The Perimeter is dead

What to trust, what not to trust?

Trust in the host

The Perimeter is dead

What to trust, what not to trust?

Trust in the host

- Attestation
- Measurement
- Hardware root-of-trust

Example: Keylime (attestation)

https://keylime.dev/

The Perimeter is dead

What to trust, what not to trust?

Hardware: for when you're really paranoid

The Perimeter is dead

What to trust, what not to trust?

Hardware: for when you're really paranoid

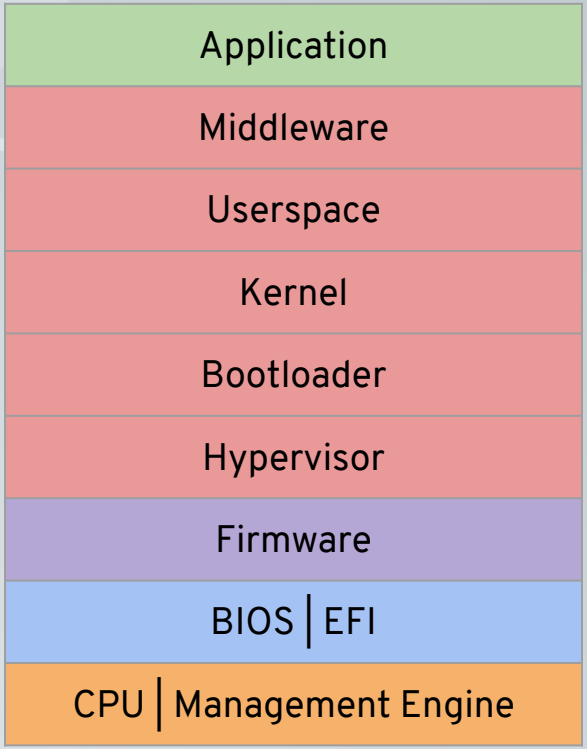
- Card-readers
- HSMs (Hardware Security Modules)
- TPMs (Trusted Platform Modules)
- TEEs (Trusted Execution Environments)

It’s DEMO time!

Announcing “Enarx” (almost)

https://github.com/enarx

Who - and what - to trust



Classic cloud virtualisation architecture

Each colour is a different trust relationship.

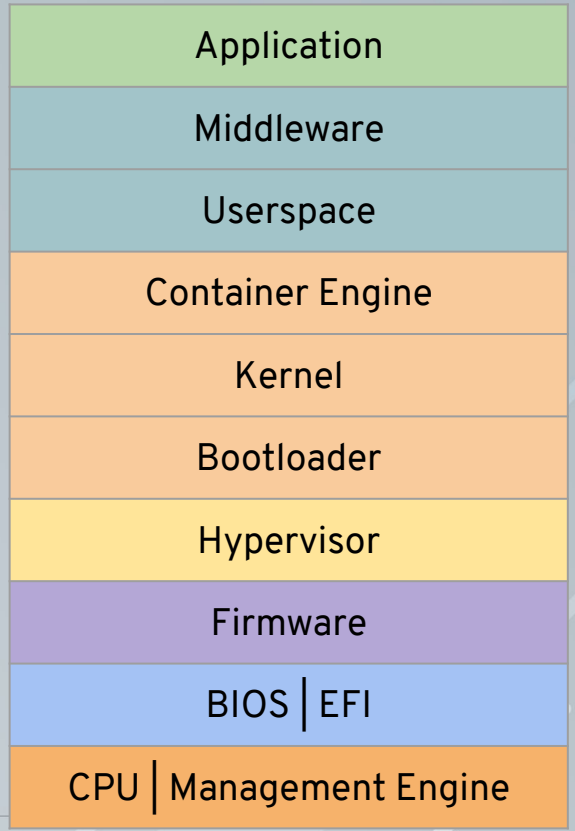
(And they *all* need to be trustworthy.)

Who - and what - to trust

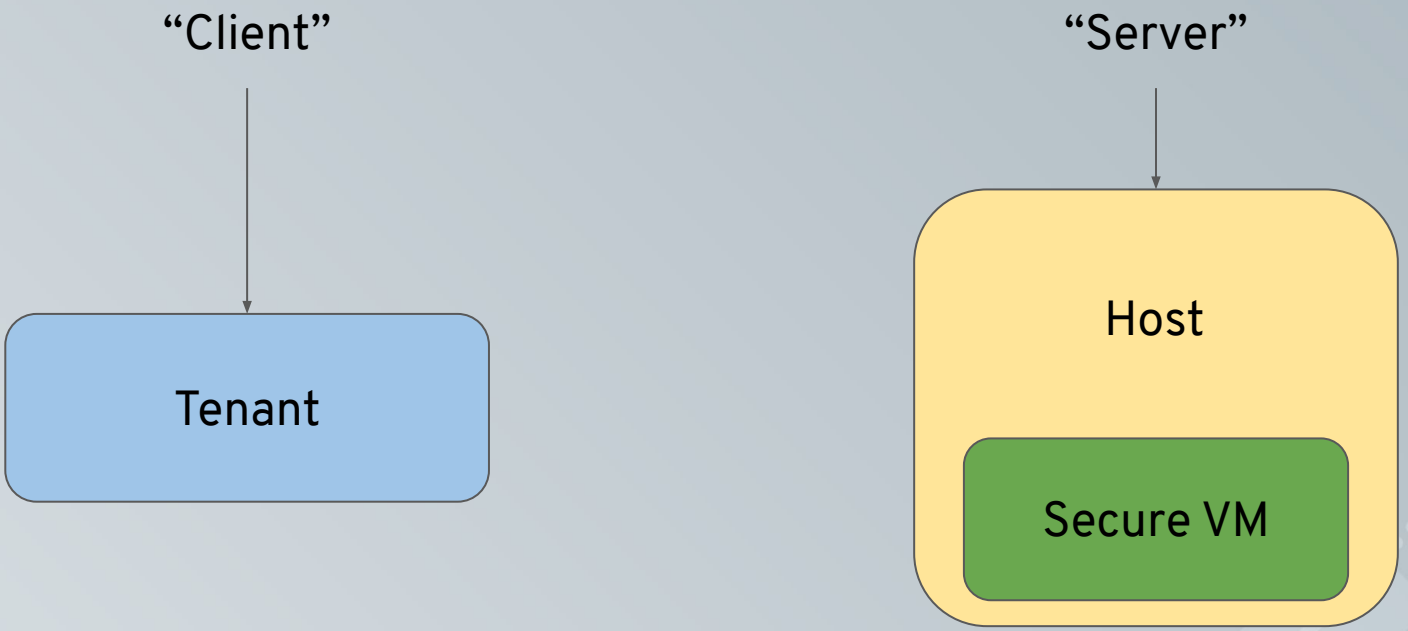
Standard cloud container architecture

Each colour is a different trust relationship.

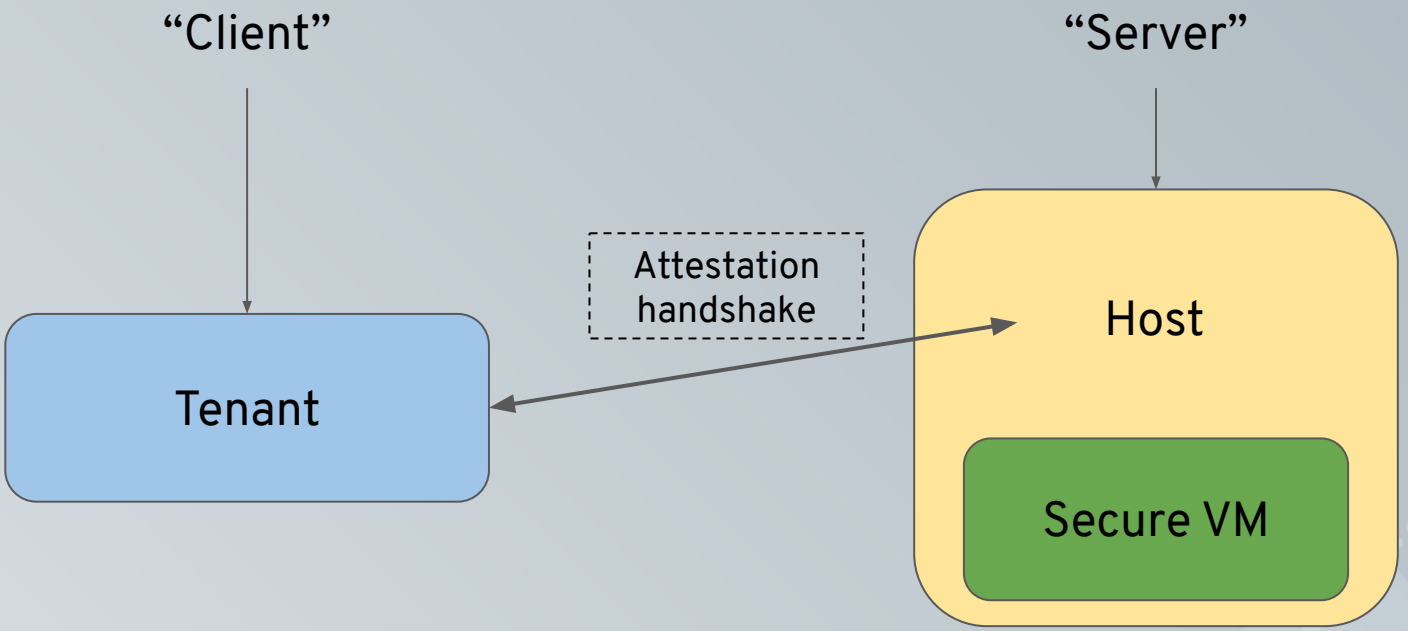
(And they *all* need to be trustworthy.)



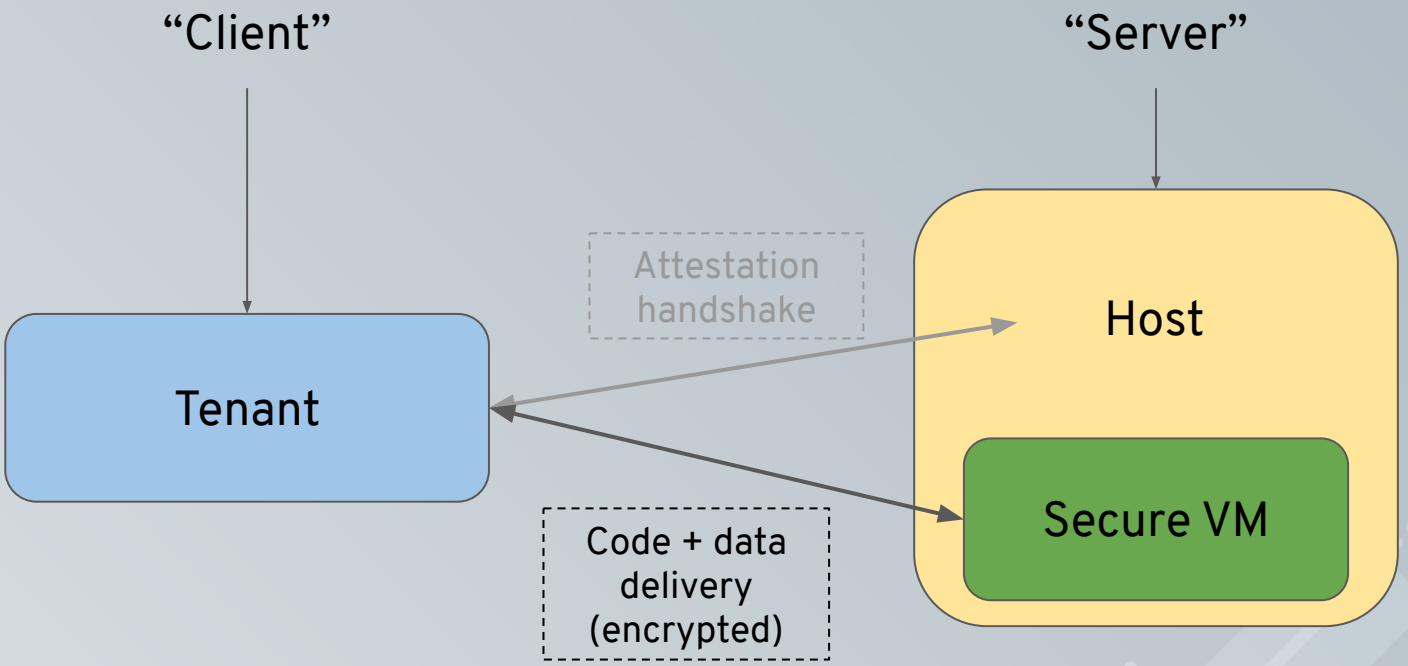
What will I see?



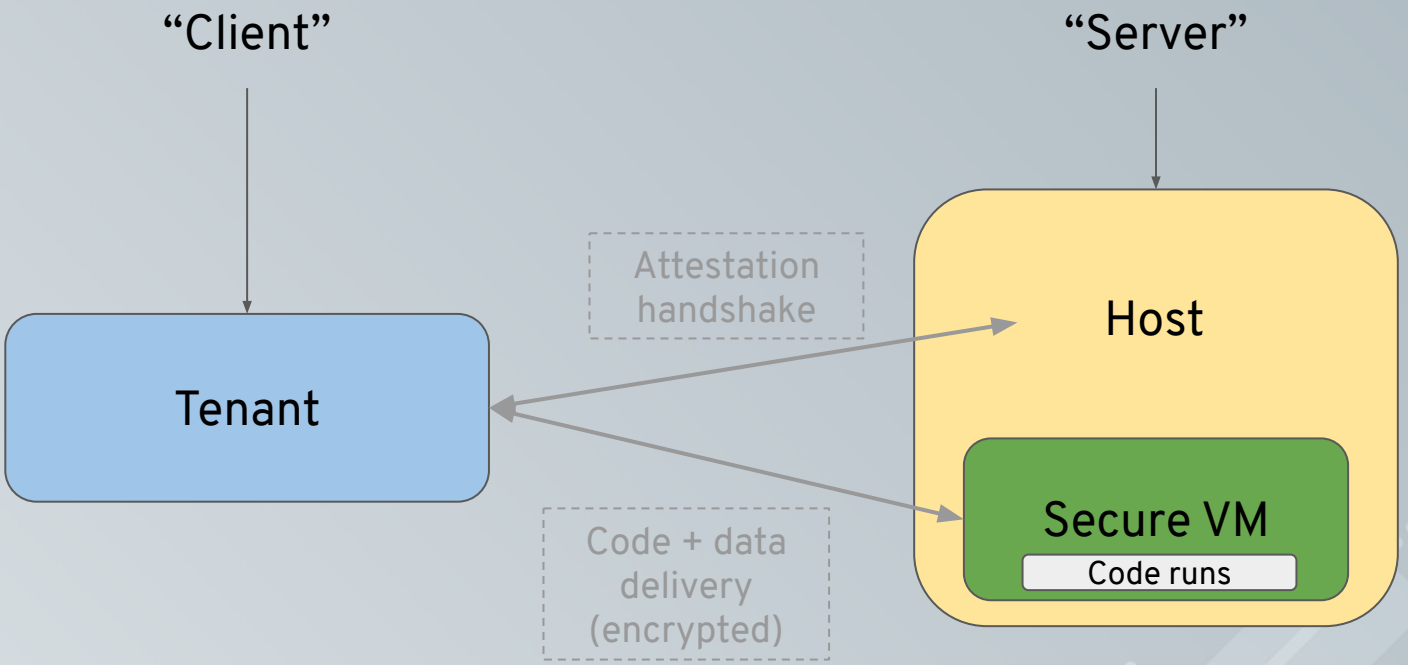
What will I see?



What will I see?

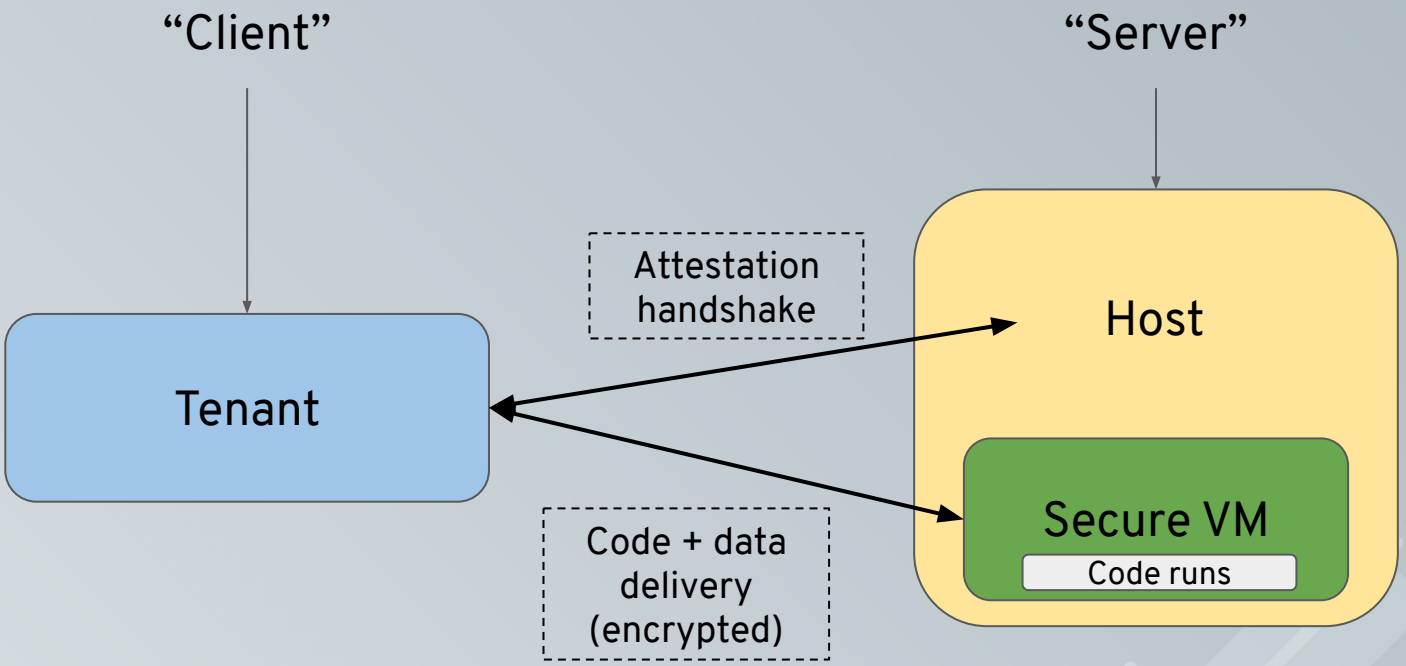


What will I see?

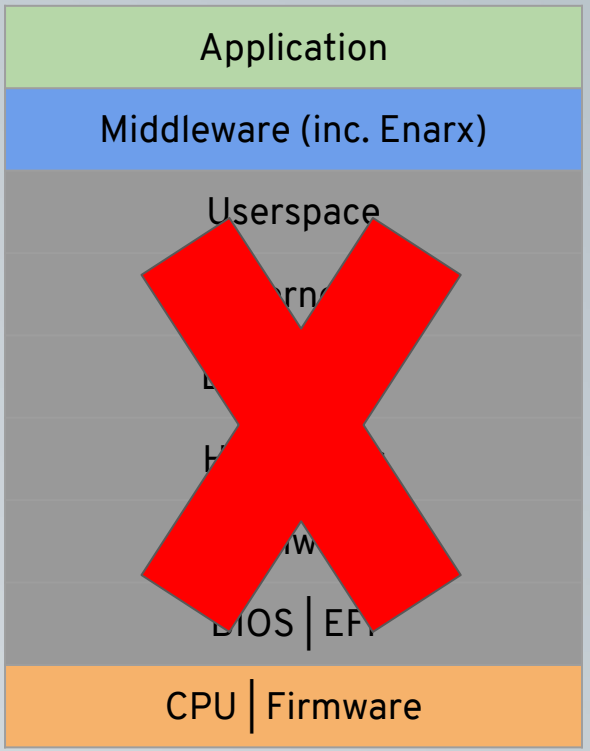


Now - the demo...

What did I just see?



Who - and what - to trust



Enarx architecture

Each colour is a different trust relationship.

Many fewer pieces.

We aim for Enarx to be small and fully auditable (and open source, of course!).

Four threat vectors

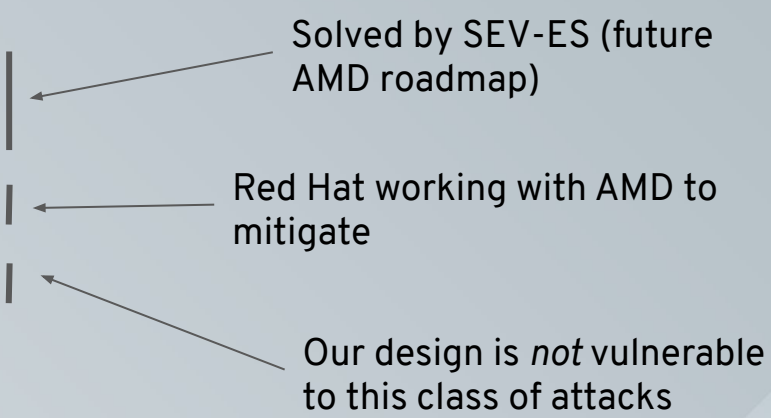
Hypervisor attacks on guest

1. Reading CPU registers
2. Writing CPU registers
3. Moving memory pages
4. Selecting decryption location

Four threat vectors

Hypervisor attacks on guest

1. Reading CPU registers
2. Writing CPU registers
3. Moving memory pages
4. Selecting decryption location



Is this all AMD-specific?

Well, it is for now…

https://github.com/enarx

Wrap up

Thank you

Enarx on github	https://github.com/enarx	
Blogs:	https://aliceevebob.com/	(Mike)
	https://npmccallum.gitlab.io/	(Nathaniel)
LinkedIn:	https://www.linkedin.com/in/mikebursell/	
Twitter:	@MikeCamel	

RED HAT SUMMIT

THANK YOU


linkedin.com/company/Red-Hat


facebook.com/RedHatinc

youtube.com/user/RedHatVideos


twitter.com/RedHat