

# The Need for Resilience

Maintaining Security When Your Hypervisor Contains Bugs

## **Intel Virtualization Security Summit**

D.J. Capelis

University of California, San Diego

<http://capelis.dj>

# The Need for Resilience

Maintaining Security When Your Hypervisor Contains Bugs

Hello folks

# The Need for Resilience

Maintaining Security When Your Hypervisor Contains Bugs

This talk has evolved

# The Need for Resilience

Maintaining Security When Your Hypervisor Contains Bugs

Originally: Casual Overview of Space

# The Need for Resilience

Maintaining Security When Your Hypervisor Contains Bugs

Now: Directly Based on My Research

# The Need for Resilience

Maintaining Security When Your Hypervisor Contains Bugs

Fortunately: Abstract works for both

# The Need for Resilience

Maintaining Security When Your Hypervisor Contains Bugs

Was it tricky writing?

# The Need for Resilience

Maintaining Security When Your Hypervisor Contains Bugs

No

# The Need for Resilience

Maintaining Security When Your Hypervisor Contains Bugs

I'm still solving the same problem

# The Need for Resilience

Maintaining Security When Your Hypervisor Contains Bugs

How do you trust buggy software?

# The Need for Resilience

Overview

Here's the overview:

# The Need for Resilience

Overview

## Software-based Techniques

The talk I originally envisioned

# The Need for Resilience

Overview

“How to design a secure hypervisor”

And why you can't do it in software alone

# The Need for Resilience

Overview

## AppSheath: A New Trusted Platform

The talk I realized I really wanted to give

# The Need for Resilience

## Overview

- **Why didn't we use TPM?**  
Sorry, I know you folks really worked hard on it...
- **The Systems Goals**  
Why did we do this?
- **Technical details**  
So what would a system like this *look like*?
- **A Few Examples**  
So what would a system like this *feel like*?
- **Implications**  
So what would a system like this *mean*?

# The Need for Resilience

Overview

Hold on a second!

# The Need for Resilience

Overview

This is the **virtualization** security summit!

# The Need for Resilience

Overview

Very true...

# The Need for Resilience

Overview

We do get back to that:

# The Need for Resilience

Overview

“How to design a secure hypervisor”

Now we can!

# The Need for Resilience

Overview

## Conclusions

Future work, for both our team and Intel

# The Need for Resilience

Software-based Techniques

So let's dive right in

# The Need for Resilience

Software-based Techniques

Software-based Techniques

# The Need for Resilience

Software-based Techniques

How do we make secure hypervisors?

# The Need for Resilience

Software-based Techniques

Steal Techniques from OS Research

# The Need for Resilience

Software-based Techniques

## Micro-kernels

Because somehow they're more suitable for hypervisors...

# The Need for Resilience

Software-based Techniques

**Basic Idea: Small Trusted Part**

Which happens to be perfect!

# The Need for Resilience

Software-based Techniques

Bottom Line: Great Idea!

# The Need for Resilience

Software-based Techniques

If anyone was willing to do it...

# The Need for Resilience

Software-based Techniques

Small hypervisors -> Small feature count

# The Need for Resilience

Software-based Techniques

Xen?

# The Need for Resilience

Software-based Techniques

Huge!

# The Need for Resilience

Software-based Techniques

VMware?

# The Need for Resilience

Software-based Techniques

Huge!

# The Need for Resilience

Software-based Techniques

KVM?

# The Need for Resilience

Software-based Techniques

I love KVM, it was 200 lines of code.

# The Need for Resilience

Software-based Techniques

Oh... and some of qemu.

# The Need for Resilience

Software-based Techniques

And Linux.

# The Need for Resilience

Software-based Techniques

There are a few very small hypervisors

No one uses them. There's a reason.

# The Need for Resilience

Software-based Techniques

Well there's other techniques, right?

# The Need for Resilience

Software-based Techniques

## Capabilities

Because coming up with a sensible capabilities model is just so easy!

# The Need for Resilience

Software-based Techniques

## Basic Idea: Separate Privilege

I'll give you my social security number... but not my driver's license number!

# The Need for Resilience

Software-based Techniques

Bottom Line: CPL -1, CPL 0, CPL 3

Pick one

# The Need for Resilience

Software-based Techniques

Everything boils down to these

# The Need for Resilience

Software-based Techniques

Either

# The Need for Resilience

Software-based Techniques

Reduce priv'd code

# The Need for Resilience

Software-based Techniques

Reduce what priv means

# The Need for Resilience

Software-based Techniques

They're not bad concepts, and we should  
use them where possible

# The Need for Resilience

Software-based Techniques

But it's not what we want

# The Need for Resilience

Software-based Techniques

They still require some trusted code

# The Need for Resilience

Software-based Techniques

Trusted code always has to be perfect

# The Need for Resilience

Software-based Techniques

Code is rarely perfect

# The Need for Resilience

Software-based Techniques

So we need something else...

# The Need for Resilience

Why not TPM?

TPM and TET, right?

# The Need for Resilience

Why not TPM?

Certainly the right crowd for it...

# The Need for Resilience

Why not TPM?

And I know you folks worked hard on it

# The Need for Resilience

Why not TPM?

But it's not quite right...

# The Need for Resilience

Why not TPM?

So let me tell you why I can't use it...

At least not all of it, as it currently exists

# The Need for Resilience

Why not TPM?

Disclaimer: I am not an expert on TPM

# The Need for Resilience

Why not TPM?

But most of you guys are

# The Need for Resilience

Why not TPM?

**You can tell me if I'm wrong**

I'm serious, I don't mind questions mid-presentation

# The Need for Resilience

Why not TPM?

Main Problem: Trusted Software Stack

# The Need for Resilience

Why not TPM?

Trusted Application
Trusted OS
Trusted Drivers
Trusted Hardware

# The Need for Resilience

Why not TPM?

TET: Trusted Hypervisor

# The Need for Resilience

Why not TPM?

Overshadow

# The Need for Resilience

Why not TPM?

XOM

# The Need for Resilience

Why not TPM?

We don't want to have to trust the hypervisor

# The Need for Resilience

Why not TPM?

Another Issue

# The Need for Resilience

Why not TPM?

TPM removes control from the end-user

# The Need for Resilience

System Goals

We wanted to create a trusted platform that  
*enabled* the end-user

# The Need for Resilience

System Goals

So our system was designed differently...

# The Need for Resilience

System Goals

**No Attestation**

This means we can't support DRM

# The Need for Resilience

System Goals

**No crypto required in hardware**

Although there's a few places it helps

# The Need for Resilience

System Goals

**User determines app's trust**

User carries around a “key” for their computer

# The Need for Resilience

System Goals

**Allows full debugging and profiling of apps**

Kind of... more later on this

# The Need for Resilience

Technical Details

## Technical Challenges

# The Need for Resilience

Technical Details

Interact with the User

# The Need for Resilience

Technical Details

Interact with the App

# The Need for Resilience

Technical Details

**Interact with the (hyper|super)visor**

Whatever is managing the real page tables on the machine

# The Need for Resilience

Technical Details

What do we need from the user?

# The Need for Resilience

Technical Details

We need to know who the user trusts

# The Need for Resilience

Technical Details

Nothing else is trusted

# The Need for Resilience

Technical Details

So how do we ask the user about this?

# The Need for Resilience

Technical Details

**User carries around a storage device**

We call it a key, but it has nothing to do with crypto

# The Need for Resilience

Technical Details

Contains pairs of the following:

# The Need for Resilience

Technical Details

A name

# The Need for Resilience

Technical Details

A bunch of binary data

# The Need for Resilience

Technical Details

Or instead, a cryptographic hash

# The Need for Resilience

Technical Details

So now the user has told us what they trust

# The Need for Resilience

Technical Details

'visors can request a trusted context

# The Need for Resilience

Technical Details

Trusted contexts are only given to things  
users say they trust

# The Need for Resilience

Technical Details

When a trusted context is given, the name  
the user gave is associate with it

# The Need for Resilience

Technical Details

Protected I/O requests in and out of this  
trusted context use this name

# The Need for Resilience

Technical Details

So now if an app asks for a password...

# The Need for Resilience

Technical Details

An LCD on the keyboard will display “firefox”

# The Need for Resilience

Technical Details

Basic Idea: Users only give their secrets to programs they trust

# The Need for Resilience

Technical Details

The 'visors can't read the secrets

# The Need for Resilience

Technical Details

But the 'visors still need to manage things!

# The Need for Resilience

Technical Details

OSs and VMMs manage resources

# The Need for Resilience

Technical Details

That's the point

# The Need for Resilience

Technical Details

## Trusted DMA Controller

Any sort of I/O controller is fine

# The Need for Resilience

Technical Details

Northbridge, on-chip....

# The Need for Resilience

Technical Details

Should be near the memory arbiter

# The Need for Resilience

Technical Details

For allocation and deallocation:

**OS sends request to trusted DMA controller**

# The Need for Resilience

Technical Details

DMA controller forces zerofilling on  
allocation of new pages

# The Need for Resilience

Technical Details

Memory arbiter controls access

# The Need for Resilience

Technical Details

Provided with following information:

# The Need for Resilience

Technical Details

## Address of PSP for Trusted Context

PSP = Program Status Page

# The Need for Resilience

Technical Details

Virtual Memory Address

# The Need for Resilience

Technical Details

Basically has another separate small set of hardware page tables

# The Need for Resilience

Technical Details

But these are shared mappings...

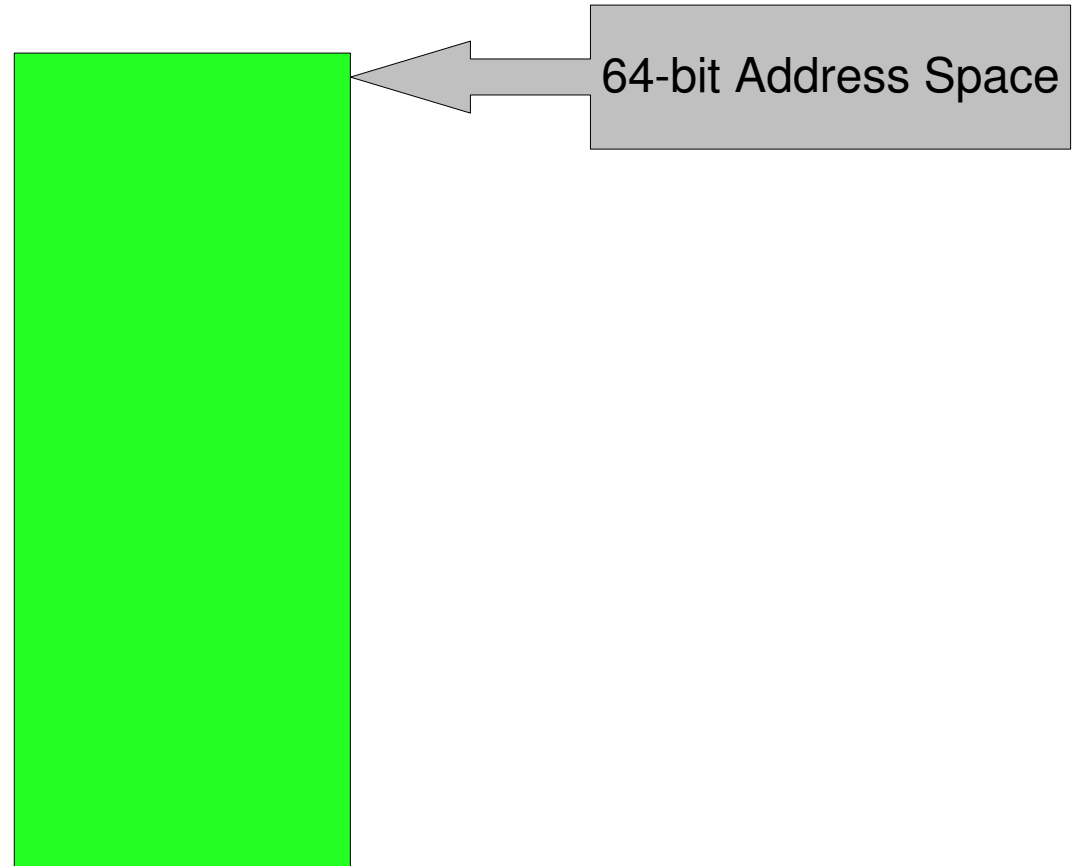
# The Need for Resilience

Technical Details

In a trusted context, memory is different

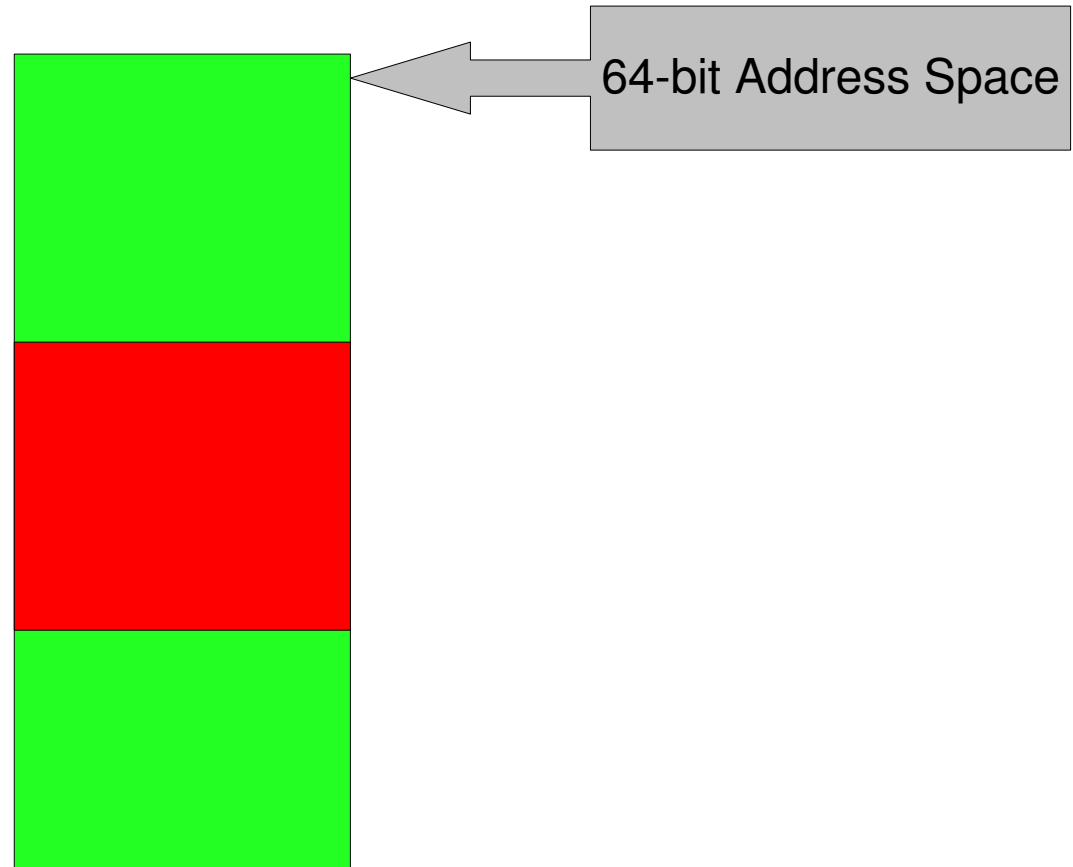
# The Need for Resilience

Technical Details



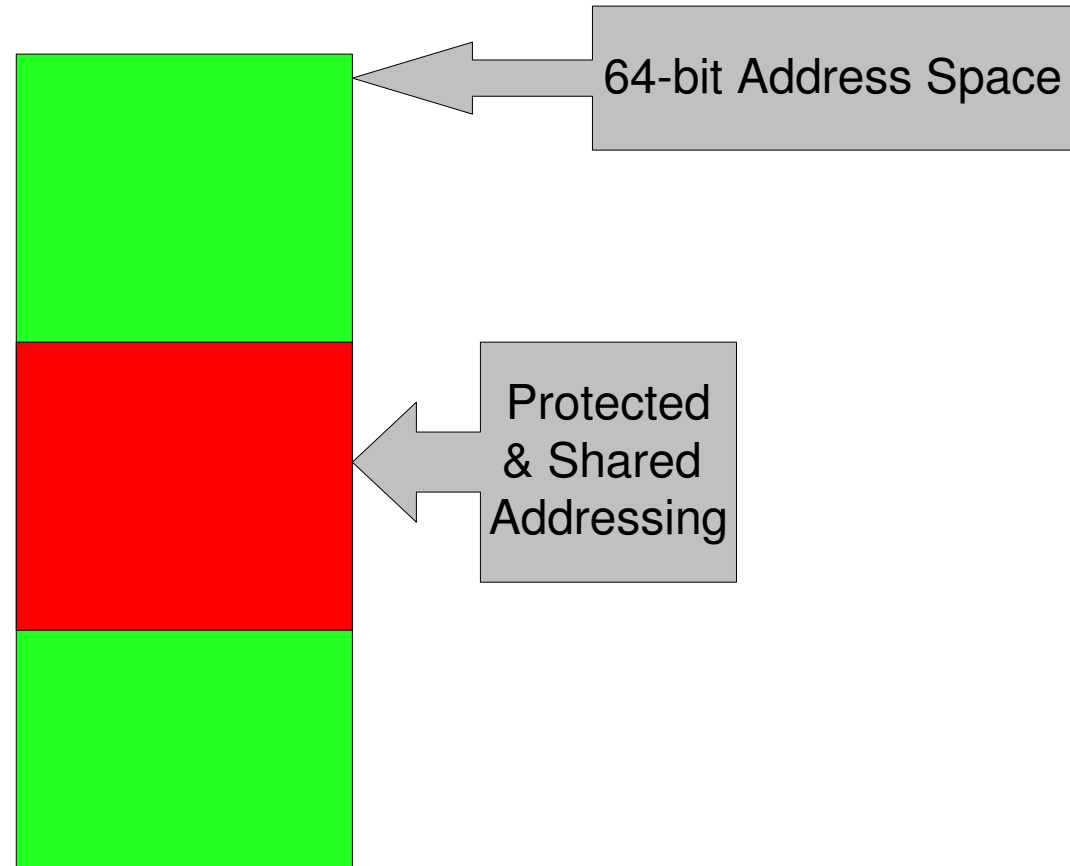
# The Need for Resilience

Technical Details



# The Need for Resilience

## Technical Details



# The Need for Resilience

Technical Details

So everything running in a trusted context  
gets the same view of memory

In that portion of the address space

# The Need for Resilience

Technical Details

The TLB is *ignored and unused*

(For things in a trusted context using this portion of the address space)

# The Need for Resilience

Technical Details

But we don't want one trusted context to be able to use another's data

# The Need for Resilience

Technical Details

**SLB is authoritative for this space**

SLB = Security Lookaside Buffer

# The Need for Resilience

Technical Details

SLB queries the arbiter

# The Need for Resilience

Technical Details

Otherwise operates entirely like a hw TLB

(And should perform about the same)

# The Need for Resilience

Technical Details

So now we have a secure place to put data

# The Need for Resilience

Technical Details

Which means we can keep secrets

# The Need for Resilience

Technical Details

Securely

# The Need for Resilience

Technical Details

Even if the 'visor is bad

# The Need for Resilience

Technical Details

A few other changes

# The Need for Resilience

Technical Details

Secrets can't stay in registers

# The Need for Resilience

Technical Details

Hardware needs to flush them

# The Need for Resilience

Technical Details

So into the PSP they go

# The Need for Resilience

Technical Details

When can a 'visor kill a program?

# The Need for Resilience

Technical Details

Whenever it wants

# The Need for Resilience

Technical Details

But when it does the hw destroys it all

# The Need for Resilience

Technical Details

No secrets get leaked

# The Need for Resilience

Technical Details

Swapping needs to still be possible

# The Need for Resilience

Technical Details

Solution: Page release mechanism

# The Need for Resilience

Technical Details

Swapping requires cooperation

# The Need for Resilience

Technical Details

OS is free to kill program if it doesn't

So a trusted program will

# The Need for Resilience

Technical Details

Can't allow jumps into secure memory

# The Need for Resilience

Technical Details

- 1) It reveals information

# The Need for Resilience

Technical Details

2) With almost all programs you can make them do anything

# The Need for Resilience

Technical Details

**Basic Attack: Jump into the middle of a block**

# The Need for Resilience

Technical Details

So we can't let things jump into secure code

# The Need for Resilience

Technical Details

So how can the OS do signals?

# The Need for Resilience

Technical Details

How can libs do callbacks?

# The Need for Resilience

Technical Details

“Requested Program Counter”

# The Need for Resilience

Technical Details

Value is pushed onto the PSP on a restore

Remember: PSP = Program Status Page

# The Need for Resilience

Technical Details

This means app can check jumps

# The Need for Resilience

Technical Details

A few other more minor changes

# The Need for Resilience

Technical Details

We're going through this quickly...

# The Need for Resilience

Technical Details

For more details, ask me for the paper

# The Need for Resilience

System Walkthrough

Let's go through an example

# The Need for Resilience

System Walkthrough

Say... with an OS running in a VMM

# The Need for Resilience

System Walkthrough

User trusts the OS

# The Need for Resilience

System Walkthrough

Doesn't trust the VMM

# The Need for Resilience

System Walkthrough

On their storage device:

# The Need for Resilience

System Walkthrough

Names OS “Linux”

# The Need for Resilience

System Walkthrough

Has hash of system image

# The Need for Resilience

System Walkthrough

VMM loads system image

# The Need for Resilience

System Walkthrough

Requests hardware provide trusted context

# The Need for Resilience

System Walkthrough

Hardware looks, hashes and agrees

# The Need for Resilience

System Walkthrough

Memory containing system image becomes  
protected

# The Need for Resilience

## System Walkthrough

Memory arbiter will deny all access until deallocation

# The Need for Resilience

System Walkthrough

**First protected page becomes PSP**

(PSP = Program Status Page)

# The Need for Resilience

System Walkthrough

'visor jumps into OS code

# The Need for Resilience

System Walkthrough

## Hardware prevents arbitrary jumps

Only prevents them from going into secure pages, otherwise it's as normal

# The Need for Resilience

System Walkthrough

Instead jumps to first address

# The Need for Resilience

## System Walkthrough

OS boots, can use secure and unsecure  
memory equally

# The Need for Resilience

System Walkthrough

Can't tell whether or not it's in trusted mode

Remember, no attestation was a goal

# The Need for Resilience

System Walkthrough

But the user can...

# The Need for Resilience

## System Walkthrough

Whenever the OS asks for input, user can see whether it's running from a trusted mode

# The Need for Resilience

## System Walkthrough

I/O works when the trusted code asks the  
'visor to do trusted I/O

# The Need for Resilience

System Walkthrough

'visor allocs protected memory

# The Need for Resilience

## System Walkthrough

'visor tells hardware to open trusted channel  
and put data into protected memory

# The Need for Resilience

System Walkthrough

If 'visor lies

# The Need for Resilience

## System Walkthrough

### 1) No trusted name on user's device

If the program isn't in trusted mode or the channel isn't properly set-up

# The Need for Resilience

## System Walkthrough

### 2) Program won't get data

At least not at a protected address if it's in trusted mode

# The Need for Resilience

System Walkthrough

Either the user will figure it out, or the  
program will

# The Need for Resilience

System Walkthrough

So they won't trust it with new secrets

# The Need for Resilience

Implications

So what are the implications of this system?

# The Need for Resilience

Implications

Anything using it must be aware of it

# The Need for Resilience

Implications

'visor must support it

(All of them)

# The Need for Resilience

Implications

User needs to carry a key

# The Need for Resilience

Implications

On the upside:

# The Need for Resilience

Implications

'visor bug doesn't mean system compromise

# The Need for Resilience

Implications

Memory protection provides basic framework for more advanced systems

# The Need for Resilience

Implications

**System works for the end-user**

So they'll trust it and want to use the security system

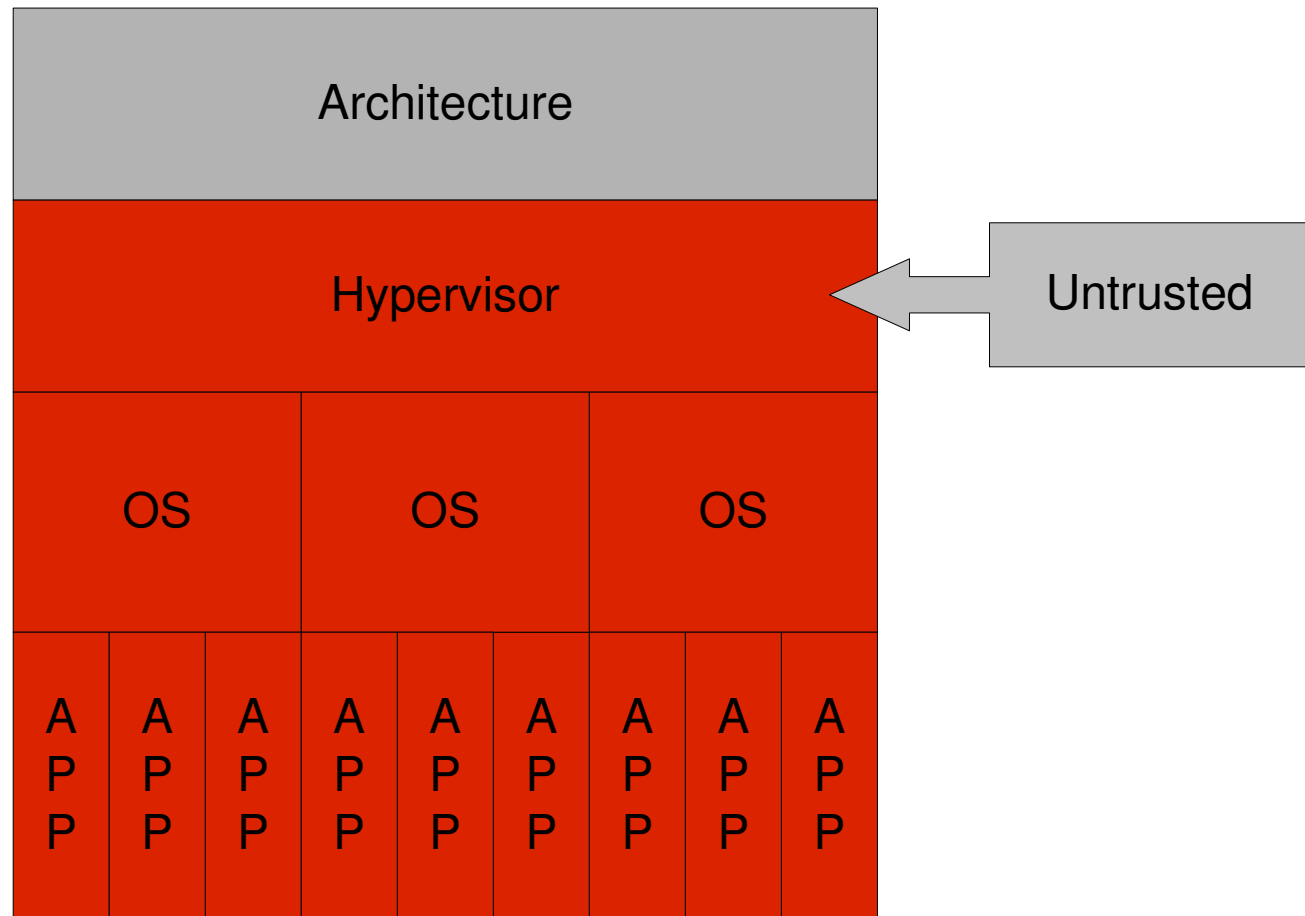
# The Need for Resilience

Applications to Virtualization

So I think you see how this applies to virtualization, right?

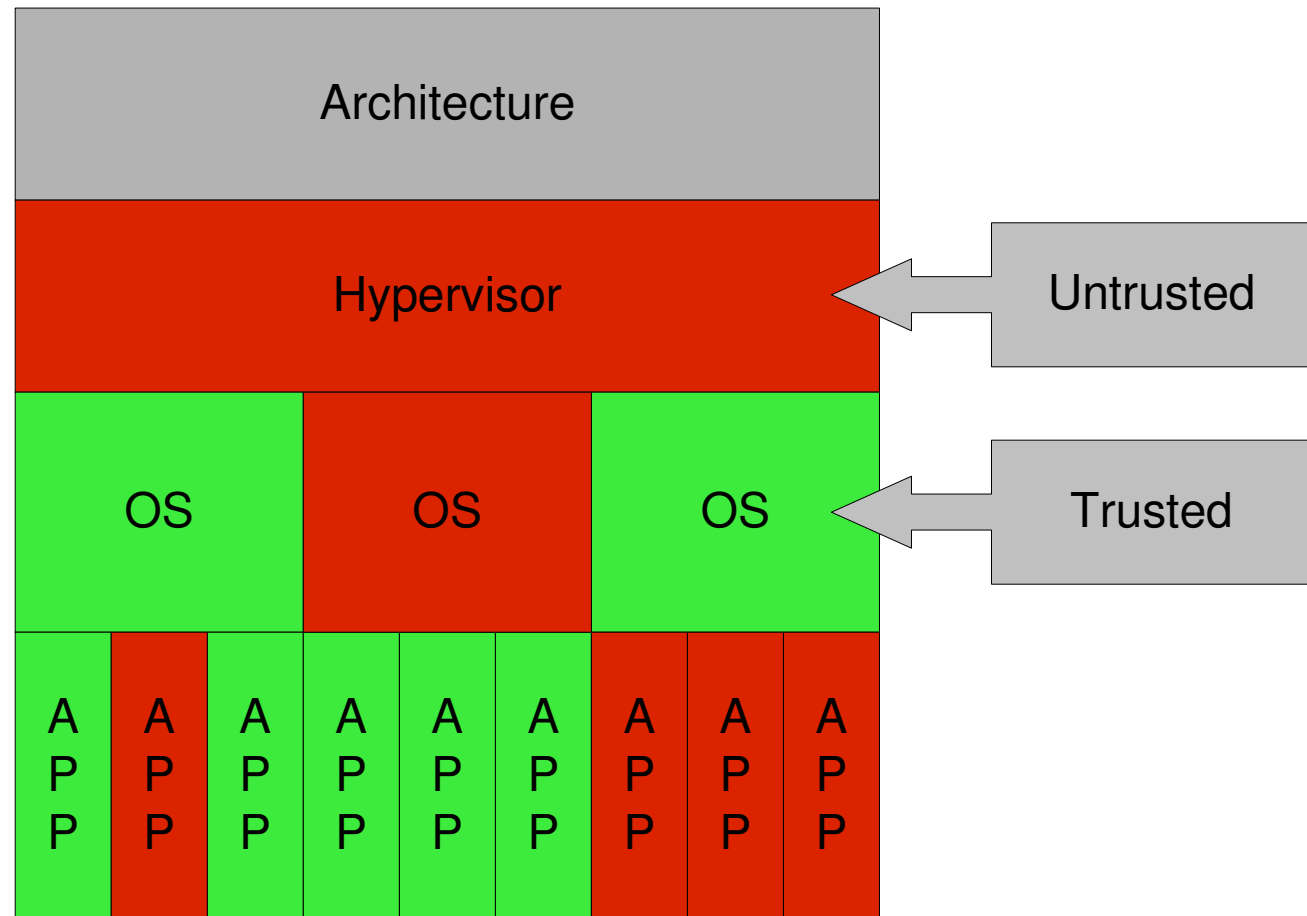
# The Need for Resilience

Applications to Virtualization



# The Need for Resilience

Applications to Virtualization



# The Need for Resilience

Conclusions

Call to Action

# The Need for Resilience

Conclusions

Virtualization security isn't that different

It's just a different visor!

# The Need for Resilience

Conclusions

Intel needs more security features in chips

# The Need for Resilience

Conclusions

We need a better security model for memory

# The Need for Resilience

Conclusions

Trusted computing wasn't a bad idea

# The Need for Resilience

Conclusions

Needs large revamp

# The Need for Resilience

Conclusions

Needs to be trusted by end-users

# The Need for Resilience

Conclusions

Security should enable, not restrict

# The Need for Resilience

Conclusions

Here's the real call to action...

# The Need for Resilience

Conclusions

Take the best parts of TPM

# The Need for Resilience

Conclusions

Combine it with our work

# The Need for Resilience

Conclusions

Create a next-gen security system

# The Need for Resilience

Conclusions

**We need it yesterday**

And if you start tomorrow it could ship in a couple years

# Questions?

[mail@capelis.dj](mailto:mail@capelis.dj)  
<http://capelis.dj>