Dan Glass

- CISO at American Airlines
- Sounds like a pretty cool job
- I mostly look at spreadsheets

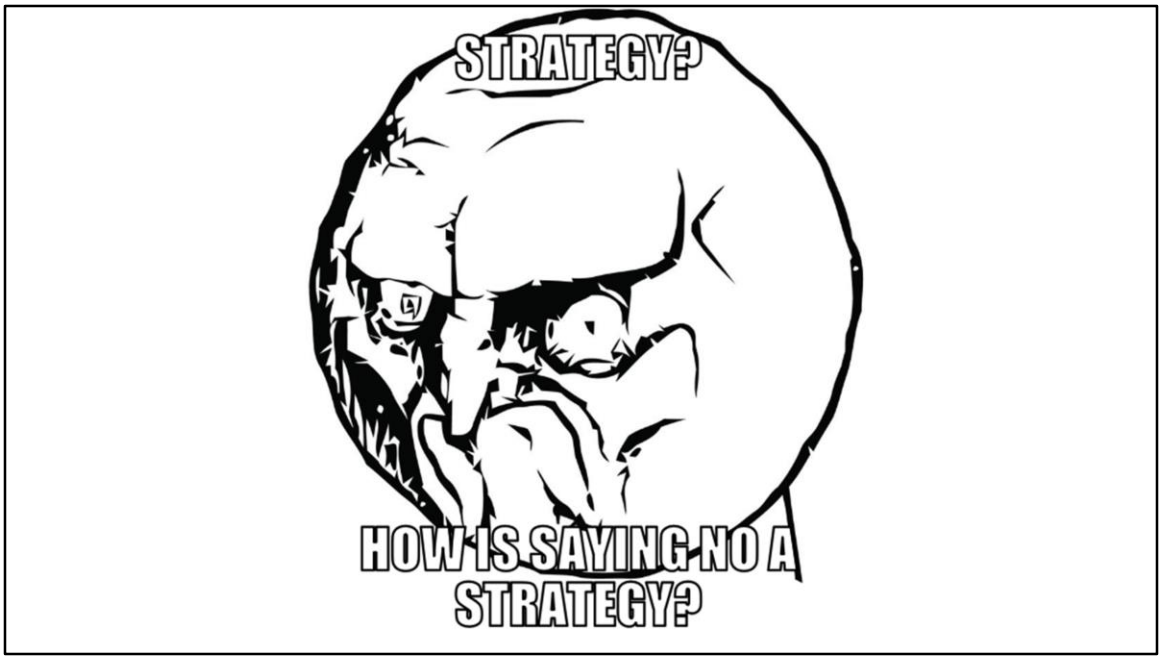
- We're pretty big
- …and there's a lot of technology that makes all that magic happen

- If you're looking to extract cool graphs, stats, or YML files… you're out of luck
- I think there's something like 5 slides that aren't memes.
- Just be thankful – I'll be done with this talk in about 30 minutes… my staff have to deal with me for 50 hours a week
- This talk has three chapters:
- Security Strategy
- Enterprise Challenges
- Opportunities (and more challenges)

- As mentioned, this is not a talk about tool chains, deployment patterns, or architectures
- Which is good because…

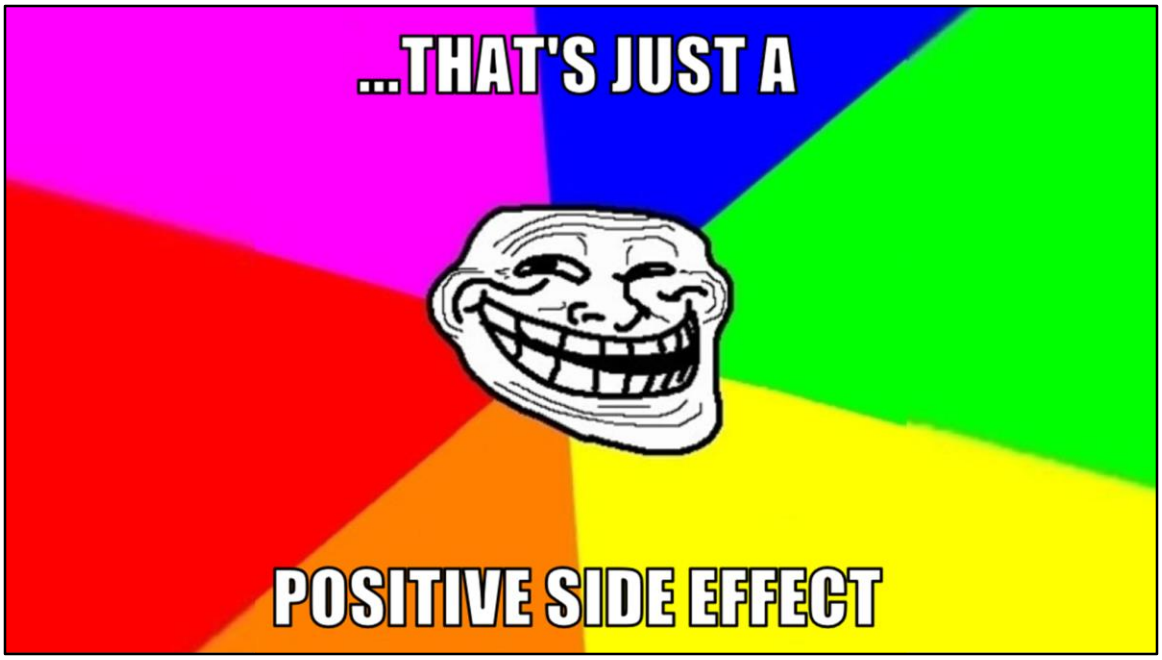
- True story

- This is a presentation how to enable a philosophy of protection that is in direct conflict with legacy controls, processes, funding, and culture
- This isn't about knife-edge transformation – this is about enabling the new mindset and functionality while maintaining legacy systems and processes
- There are definitely some things that make airlines unique – but also some that make them just like any other large complex enterprise



- I will share my overall Information Security strategy and the path we're taking to balance new methods of deployment with the traditional ways of doing things

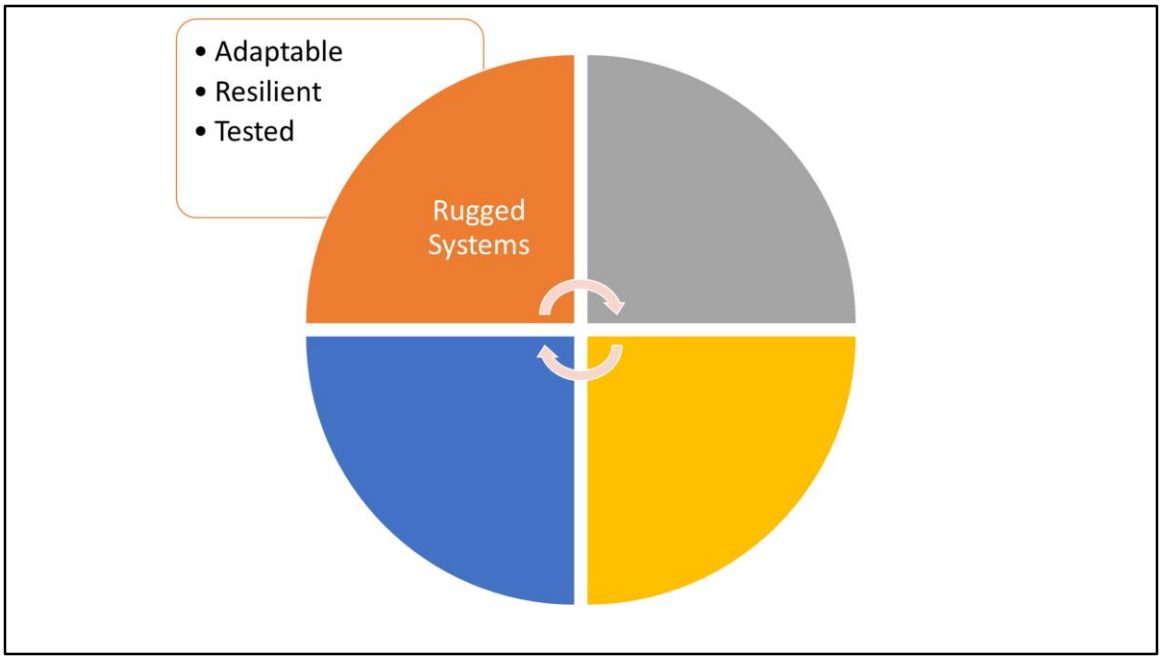
- I talk to a lot of my peers across many industry verticals – the challenges of running a large and complex IT organization are not unique to my company or aviation
- If you work for a legacy company large or small you'll recognize some of these challenges



- That being said, before we can discuss strategy we need to be able to deliver one…
- Need a talented team of professionals
- That believe in what they do and who they do it for
- Leadership smart enough to know when to allow those talented team members to go do amazing things

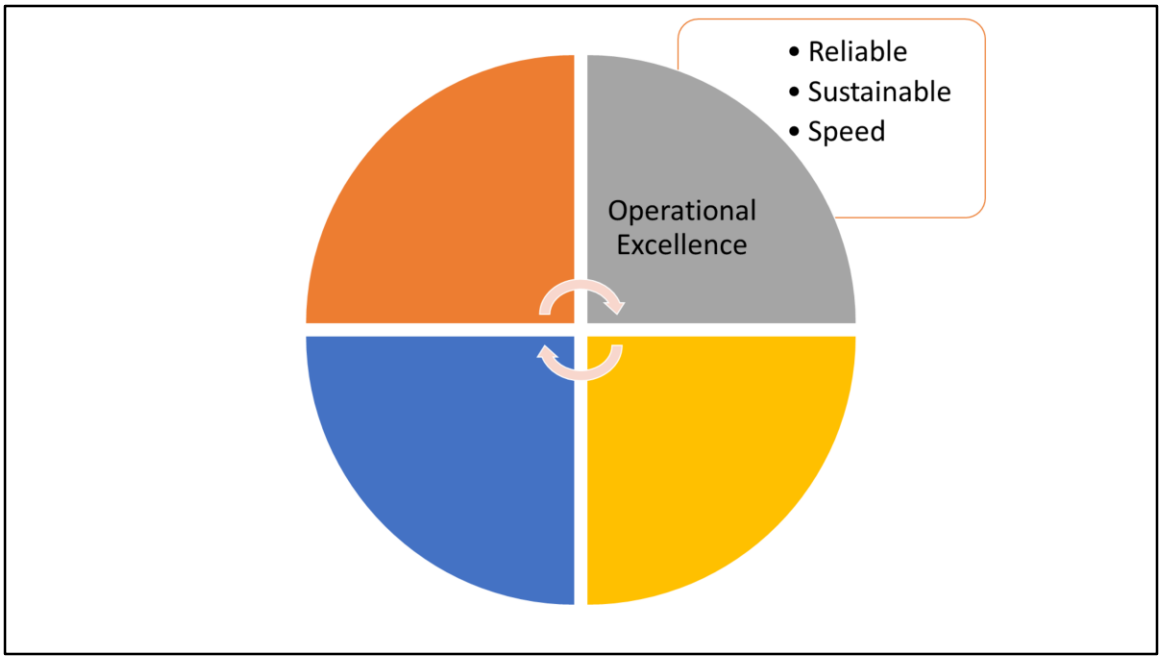
- Focuses on the basics of defense
- It doesn't depend on any specific technology, architecture, or process
- Strategy easy to understand and durable
- "control du jour" will easily fit within the strategy
- carefully chosen words
- Because words matter

- Everything in an airline has to be an acronym
- ROAD
- Rugged Systems
- Operational Excellence
- Actionable Intelligence
- Defensible Platforms
- I will describe each briefly and give some examples of programs and services we use to achieve the strategy

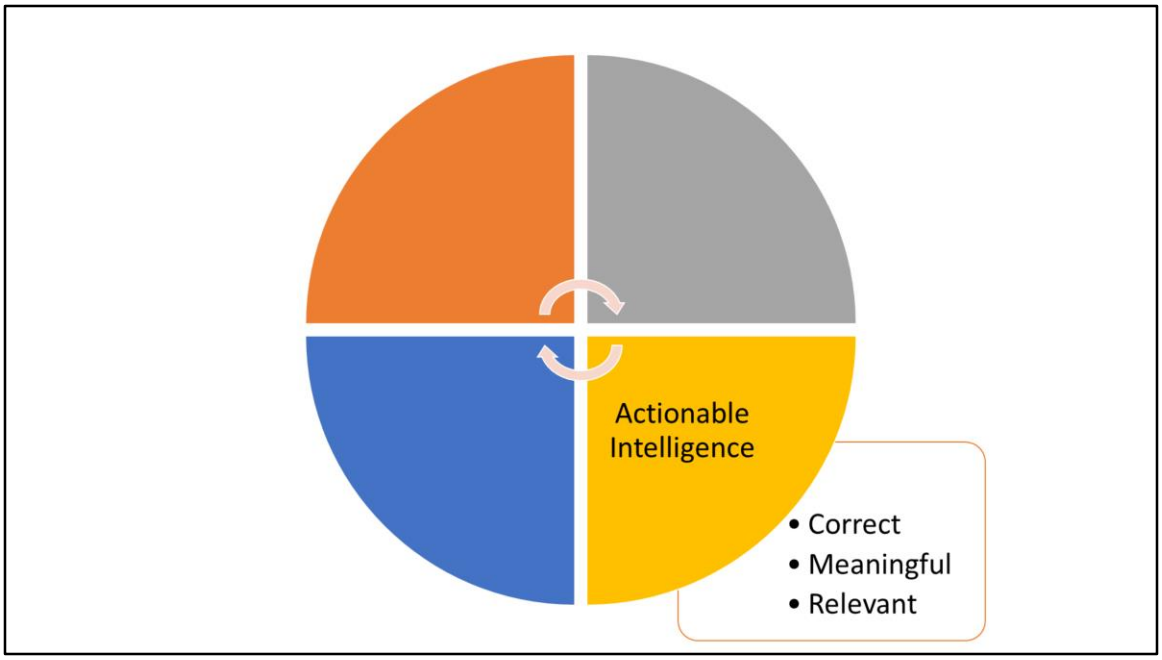


- We will aid the business to develop and maintain RUGGED SYSTEMS that can survive in hostile environments.
- We will design these systems to be RESILIENT against attack.
- We will ensure systems are ADAPTABLE to changes in environment.
- We will ensure systems are TESTED repeatedly to ensure they meet enterprise standards.
- Because the answer to any security question shouldn't be "moar firewalls!"
- First thing you need is an application security program that features:
- Secure Coding standards
- Automated code testing / analysis / reporting
- Developer training / outreach / consulting
- Build systems that are secure from the ground-up
- Architectural guidance is key
- Produce security "blueprints" for functional requirements
- Help enable DevOps tool chain (automated scanning) but still support legacy SDLC, older programming languages, manual processes
- The standards, testing, training must be flexible to work with your entire organization
- Outreach to developers, management - understand focus, pressures, how Rugged

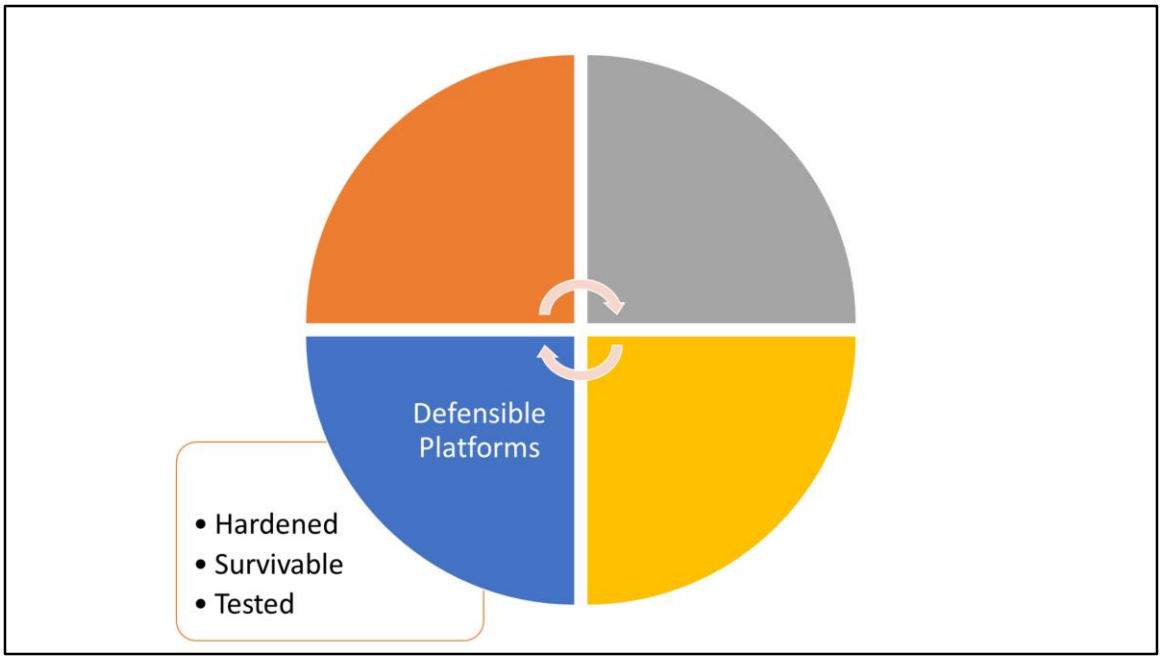
fit into their lifecycle
- Sell code quality, fewer defects, and make presence in their life as painless as possible
- Be an ally not a roadblock



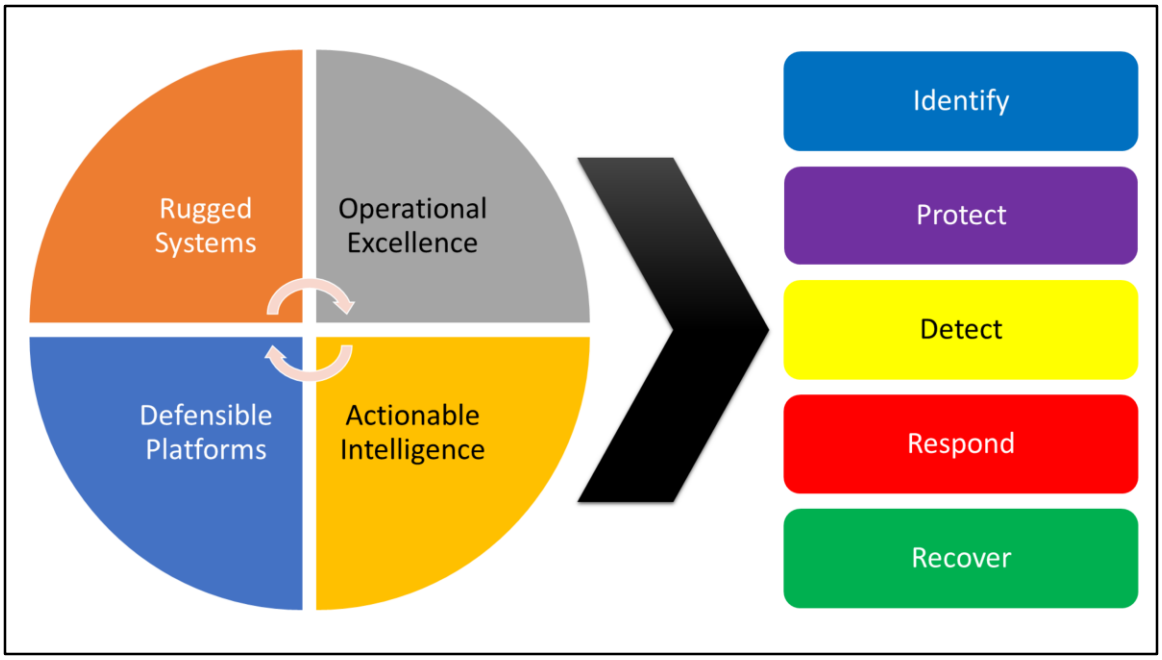
- We will strive to maintain OPERATIONAL EXCELLENCE to meet the increasing demands of a dynamic organization.
- We will build engagement and delivery processes to ensure RELIABLE services.
- We will standardize procedures across services to create a SUSTAINABLE environment.
- We will increase usage of automation and leveraged services to increase SPEED to market.
- All about process
- Centralized or federated, manual or automated
- ingrained into culture or goal will be unobtainable
- This is about those “blocking and tackling” tasks that are vital to the success of any information security and risk management program
- Boring but important
- Things like patching (yes, that’s still a thing)
- Asset/configuration management
- Change management/detection
- Toughest parts of IT budget to defend and maintain



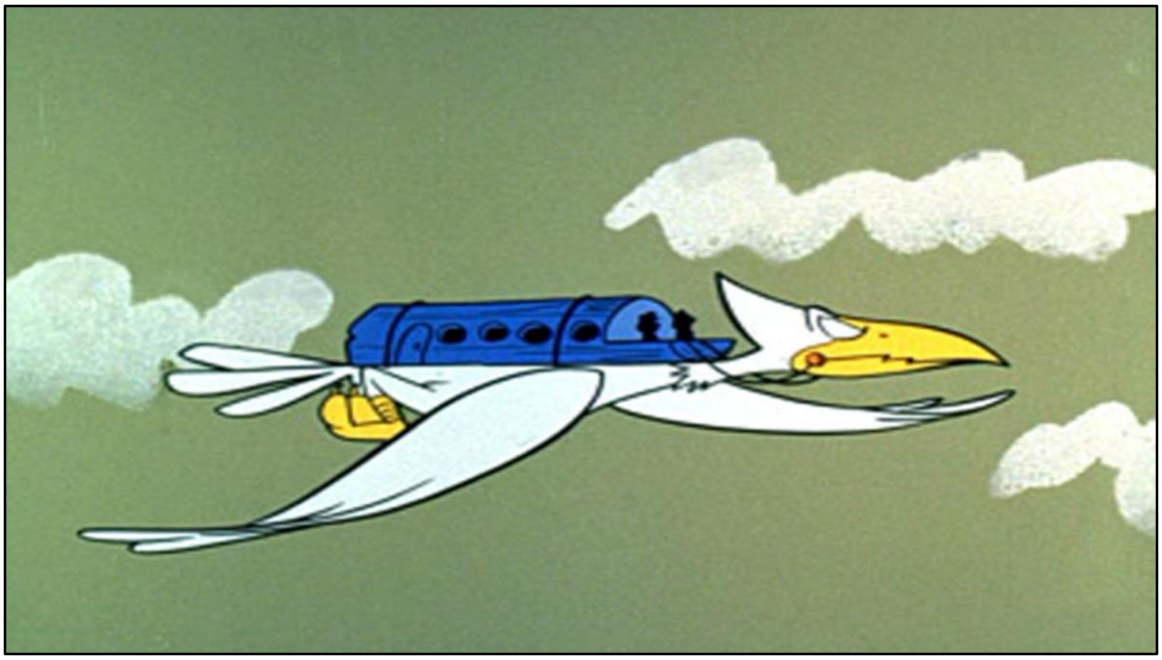
- We will harvest, analyze, and rapidly convert large sets of data into ACTIONABLE INTELLIGENCE to enable quick decision-making.
- We will perform regular data source hygiene to ensure the data are CORRECT.
- We will ensure that data are MEANINGFUL by collecting the right data from the right systems.
- We will perform analysis to ensure only RELEVANT information is communicated.
- Visibility
- running a 24x7x365 SOC
- outsourcing to a third party
- partnering with IT peers to share their
- Maya story
- Turning on the lights is not enough
- invest in platforms that let you cull through vast amounts of data
- invest in people to understand context and act on it appropriately
- People > Tools



- We will develop, deliver, and maintain DEFENSIBLE PLATFORMS that protect enterprise information resources.
- We will develop common standards to ensure all platforms are HARDENED.
- We will design our platforms to SURVIVE against sustained attack.
- We will enable business activity by deploying EFFECTIVE defenses to mitigate malicious activity.
- This is where folks like me are most comfortable
- Design platforms secure from the ground up
- Solid security architecture framework is vital to this mission
- Controls are being applied as consistently as possible
- Many layers of controls throughout network / ecosystem
- Network
- Platform
- Identity
- Access

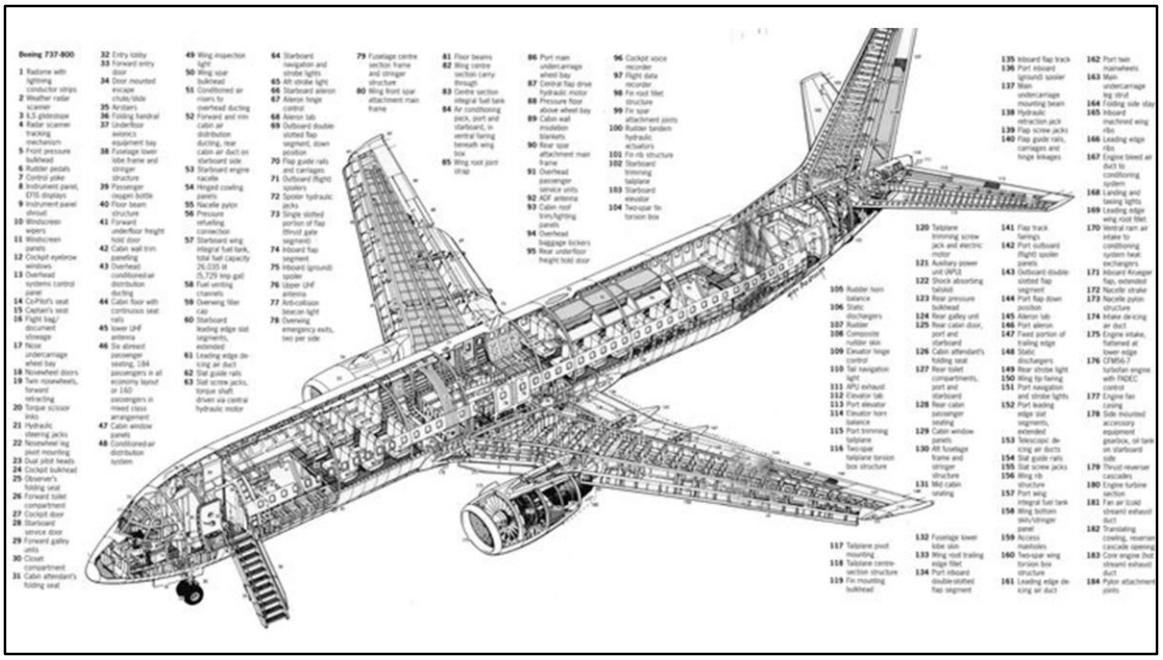


- The strategy covers all aspects of an information security program
- It even aligns to the NIST CSF (whew!)
- Identify
- Protect
- Detect
- Respond
- Recover



- What are some of the airline-specific challenges?
- There is the aircraft itself
- Long half-life
- Aircraft can be in service over 30 years

- Here you can see inside one of our cockpits
- Onboard IT systems are updated often and can be replaced but at great cost to airline
- Therefore, design issues that require modifying hardware are problematic – measure twice cut once is a mantra we repeat often when referring to systems that touch aircraft
- There are manual processes that deploy software to digitally enabled aircraft
- These processes rely heavily on separation of duties, integrity checks, and physical access – and they work pretty darn well
- The concept of safety is ingrained deep into these processes and technologies
- In other words, these processes dictate the flow of updates from manufacturers, integrators, and internal IT to the aircraft



- Sometimes slow is necessary and even desired – the concept of “fail fast” doesn’t really work in this case
- It is incredibly expensive for the airline to make changes to an aircraft so utmost care must be given to ensure we minimize impact to the fleet
- For example, when we want to update something like a certificate revocation list we have to either pull an aircraft out of service or wait until it’s scheduled “deep” maintenance check (which can be months off)
- So, therefore if there was an error in the update we would in effect ground our entire e-enabled fleet until maintenance & engineering had a chance to touch each and every aircraft before they could be entered back into service
- If I sound repetitive it’s because of how important this point is – mistakes are extremely costly and disruptive

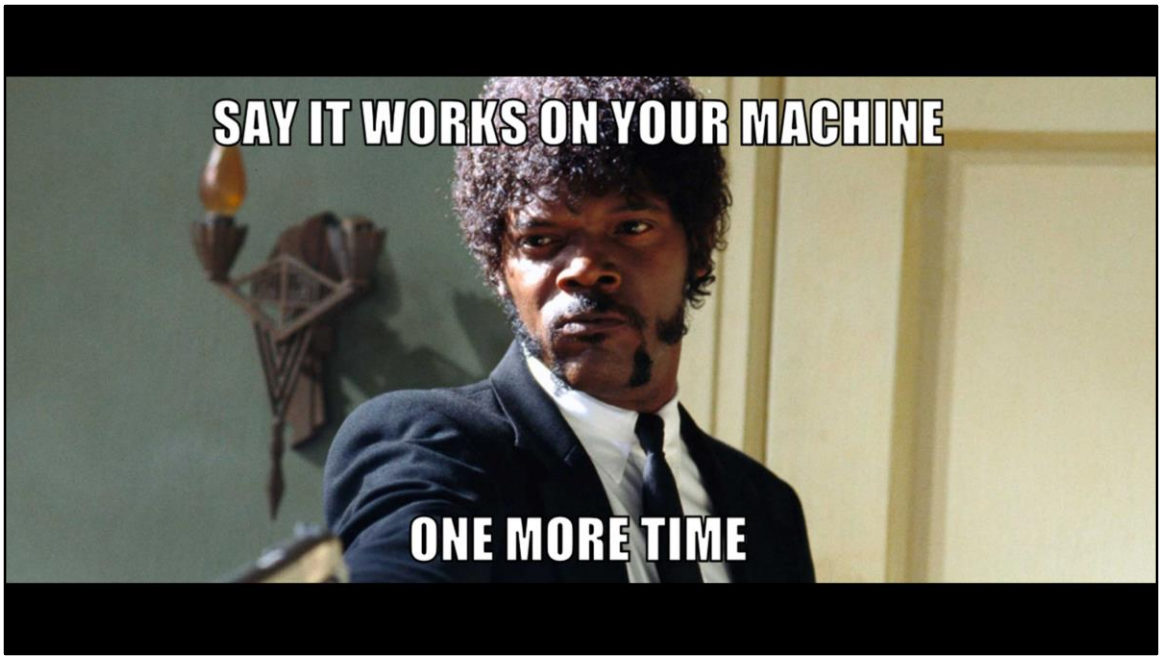
- Now let's talk about some of the issues we face in a large complex enterprise
- This is not one of our new planes – but a boy can dream…

- IT groups align to business units
- Business units have unique needs that require unique solutions and expertise - one size doesn't fit all
- This creates smokestacks that silo expertise, systems, and delivery away from each other
- This is efficient for the needs of the business - but creates issues for central enterprise services like security
- One of the negative aspects of smokestacks is that knowledge is often trapped in chimney
- Common practices, domain knowledge, and lessons learned aren't communicated

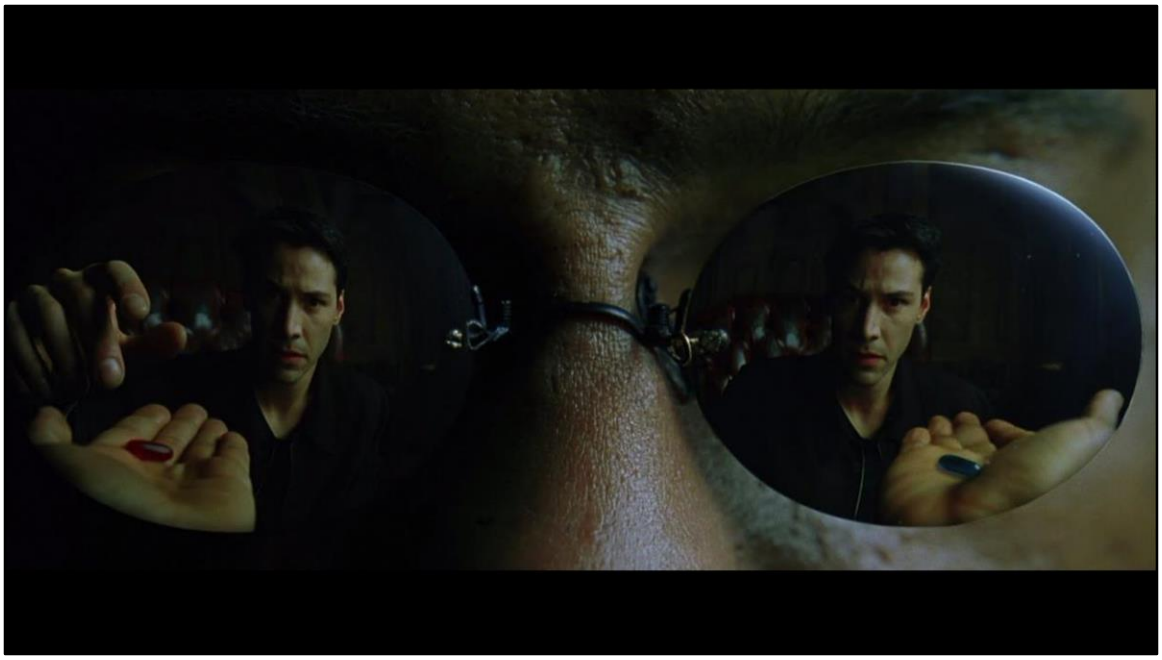
- I know Brooklyn thinks it has the market cornered on “hipster”
- But we have artisanal, local, organic, silicon-based systems that are hand-crafted with love for each deployment and are built to last
- So, even with security baselines, configuration standards, and build standards systems are still hand-crafted with love
- Virtualization and cloud were supposed to fix this problem
- However, these technologies bring raw capabilities but lack the processes and culture to make them effective by themselves

- Literally thousands of systems, subsystems, and processes run an airline day-to-day
- These systems have high levels of dependence on each other
- This can occur via hardcoding parameters or locking in on a certain version of vendor software
- We end up with systems with thousands of hard dependencies both upstream and downstream
- As you can guess the stack becomes extremely brittle and so you hear “OMFG DON’T TOUCH ANYTHING!”
- This issue slows planning, development, testing, and deployment

- Many systems have been built over the years that do their job perfectly well and don't change much
- Core airline systems pre-date TCP/IP and Nixon as POTUS
- Obviously many things have been modernized (but we still have TELETYPE)
- We also have over 1,100 modern applications that take the data from those core mainframe systems and make magic happen every day

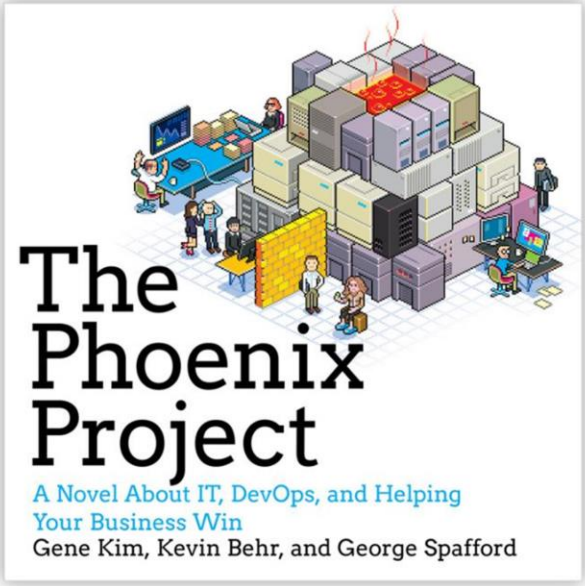


- None of these systems make a company inherently insecure or unstable
- In some limited cases these problems can be seen as a benefit - processes that ensure errors are eliminated before they get into a production system
- However, they do make them slow to deliver new functionality, fix problems, or pivot in new directions



- Security is ready to take the red pill
- DevOps solves some of the issues that legacy IT faces
- Automated configuration, testing, and deployment would help the airline get its products and services to customers and employees faster
- Immutable infrastructure is awesome
- Automated security scanning? Bitchin'
- Nobody gets interactive access to prod? OMG WHERE DO I SIGN UP?
- But all this still creates many new... ahem "opportunities" for infosec
- We still have long lead times and tons of WIP due to rigid legacy operational processes, tightly coupled systems, and more neck than bottle at times
- An application team may have 500 developers all wanting to push code but if the one gal that knows the build process inside and out goes on vacation…
- And we still have those pesky firewalls…

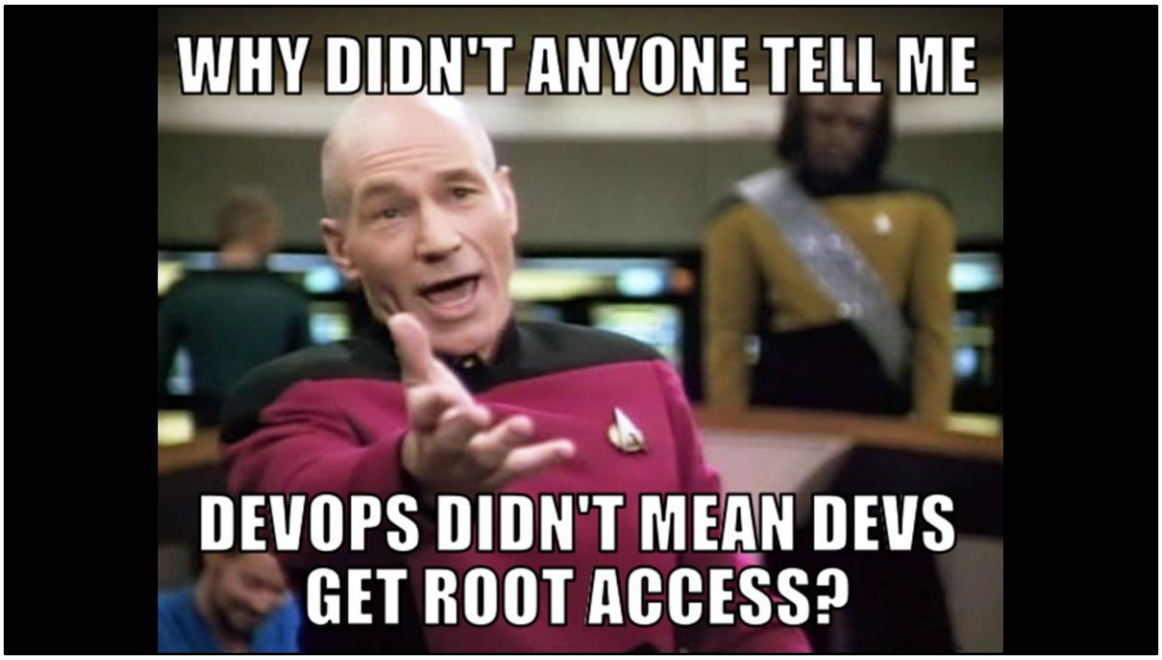
- But it's such a different paradigm it's tough to explain to team
- Where do we start?
- Lean? It’s a lot of work to explain how we can apply Toyota’s manufacturing method to what we do – everyone is busy and not everyone reads 2-3 books a month
- Agile? Mature agile teams at AA but not everywhere due to organizational and cultural challenges – as will be explained later
- DevOps? If you don’t understand Lean or Agile DevOps just looks like a set of tools to wrap manual processes and oversight to
- In other words – if you don’t get why lean thinking, agile development, and devops delivery works… you’re doomed to fail



- Phoenix Project is a great start (it's the book that started my journey a few years ago followed quickly by "The Goal" and "Continuous Delivery"
- You must be ready to follow it up with discussion and support to change things – otherwise people will remember the first half (all the problems) but forget the lessons learned in the second half (Lean, Agile, DevOps).
- Requires leadership buy-in to ensure culture change

- To drive the point home and give us some perspective about how this ain't easy: Netflix recently shared that they finally migrated 100% of their legacy operations to AWS
- It took Netflix 7 years to get there. I have no doubt they could have done it sooner if they didn't have a business to grow and run
- Most IT organizations don't have Netflix's business model, flexibility, or commitment to the model – this "all in" approach is awe-inspiring but if it took Netflix 7 years… how long will it take us?
- So the challenge is:
- We want to enable/encourage DevOps culture, processes, and tools
- We must maintain/improve legacy protection
- We must continually adapt to emerging threats
- Oh and we have to keep costs in check – we don't all get unlimited budgets

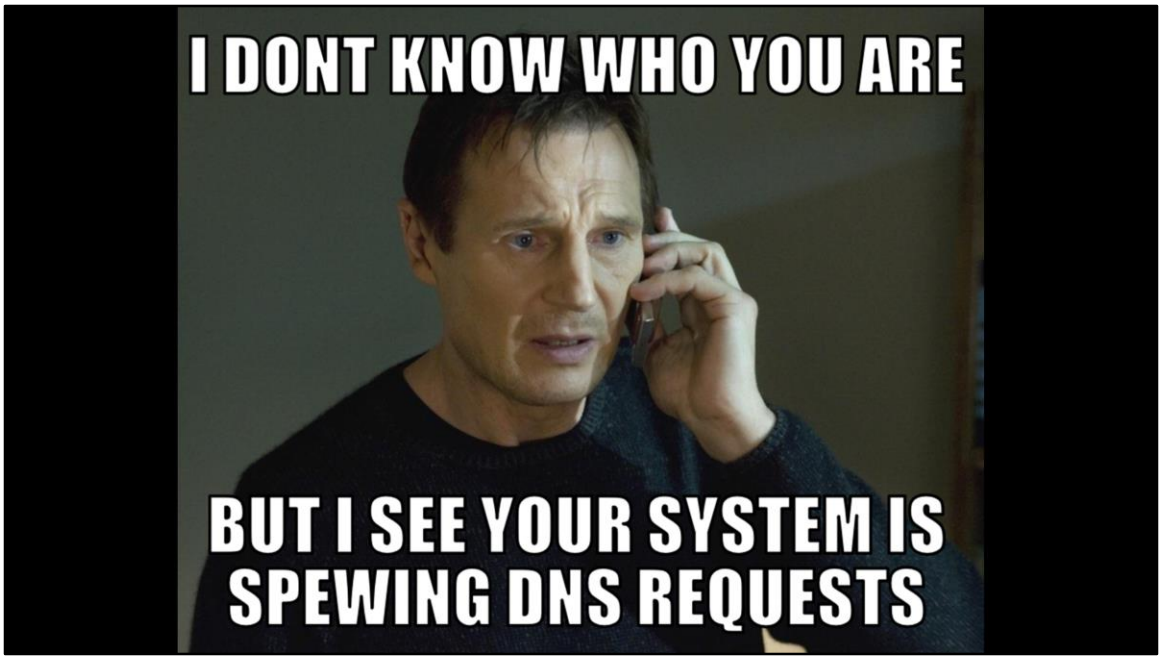
- Basically we want magic
- But I believe in magic ☺



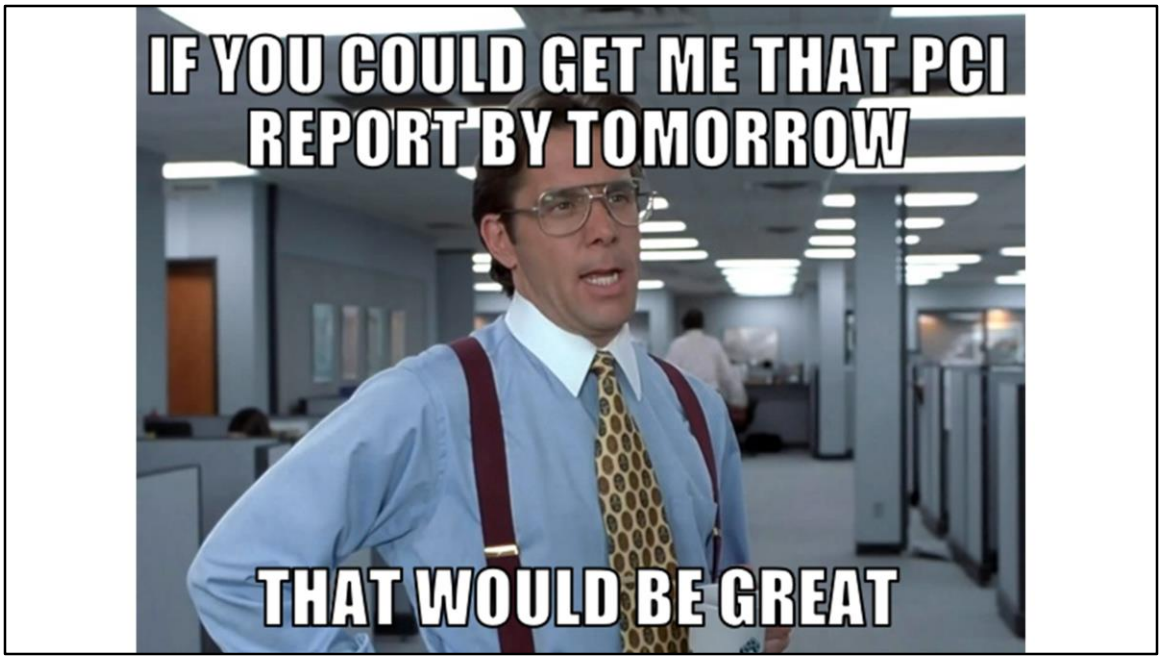
- Embrace DevOps but make sure to keep giving legacy a squeeze now and then
- New is exciting and sexy but if you don’t account for the existing systems/issues you’ll find yourself with decaying infrastructure and you’ll be back to fighting fires instead of delivering value (remember Netflix took 7 years!)
- In other words, don't replace one set of problems with another whole new set of worse problems

- Automate what you can; solid process for the rest
- Encourage automation but understand that many see automation as a threat to their livelihood so prepare to meet resistance here
- This is probably the toughest part of the culture shift outside the whole “it’s new and different” thing

- Automated controls increase security, accountability, and consistency
- Therefore, certain processes and controls can be shed if devops is adopted



- DevOps toolchain can increase visibility
- This visibility can ensure ops/secops can jump on problems quickly



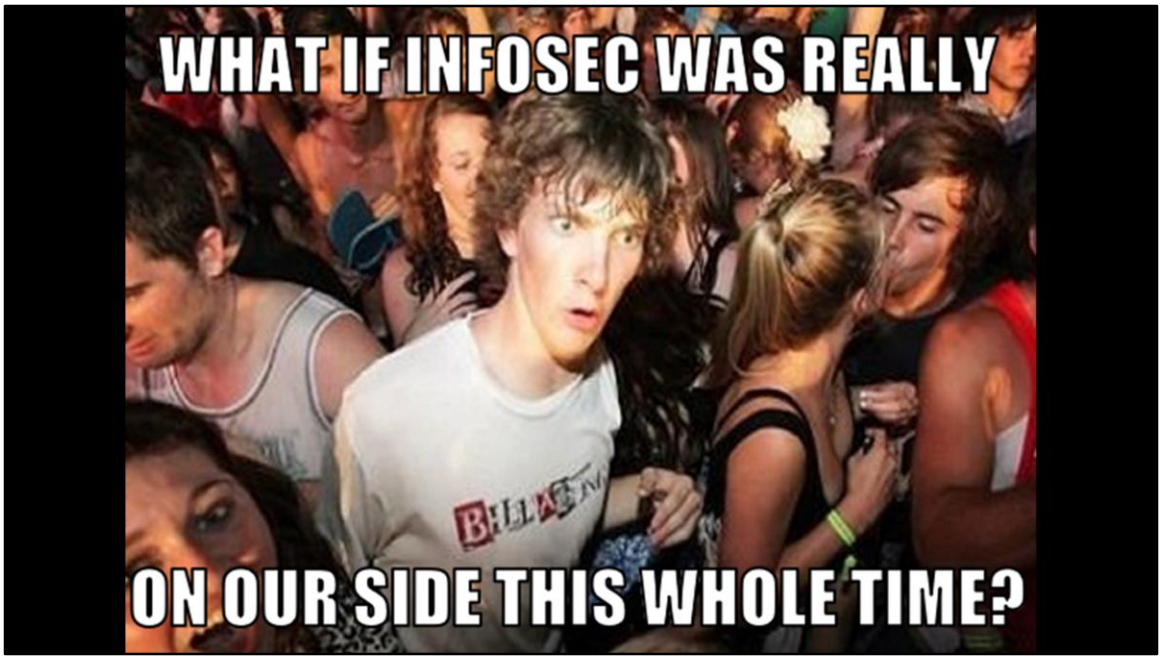
- Remember, compliance is still a thing
- Explaining devsecops to a QSA or auditor isn't as easy as you think
- Eventually they nod their heads
- Winning!!!
- Then they'll say
- "but PCI DSS x.x.x says and I quote..."

- Culture lets us experiment with new tools, tools enables process improvements, process improvements improves the culture, culture enables further experimentation in tooling
- Without all three you're doomed to repeat past behavior

- Be wary here – let me take a moment to recount how to make something cool suck
- When we (AA) first deployed to cloud in early 2010
- Security was actually a champion of the concept and pushed for it - the nirvana you all are achieving today
- However, when we designed the security controls we layered in cool bleeding edge security tech but continued to use our legacy processes
- Therefore, we ended up with a bunch of systems in the “cloud” but in reality it was just another data center deployment from a speed perspective (outside spinning up the original VM)
- Obviously this was a long time ago and we’ve matured
- But the lesson still stands out as we approach devops and whatever is next

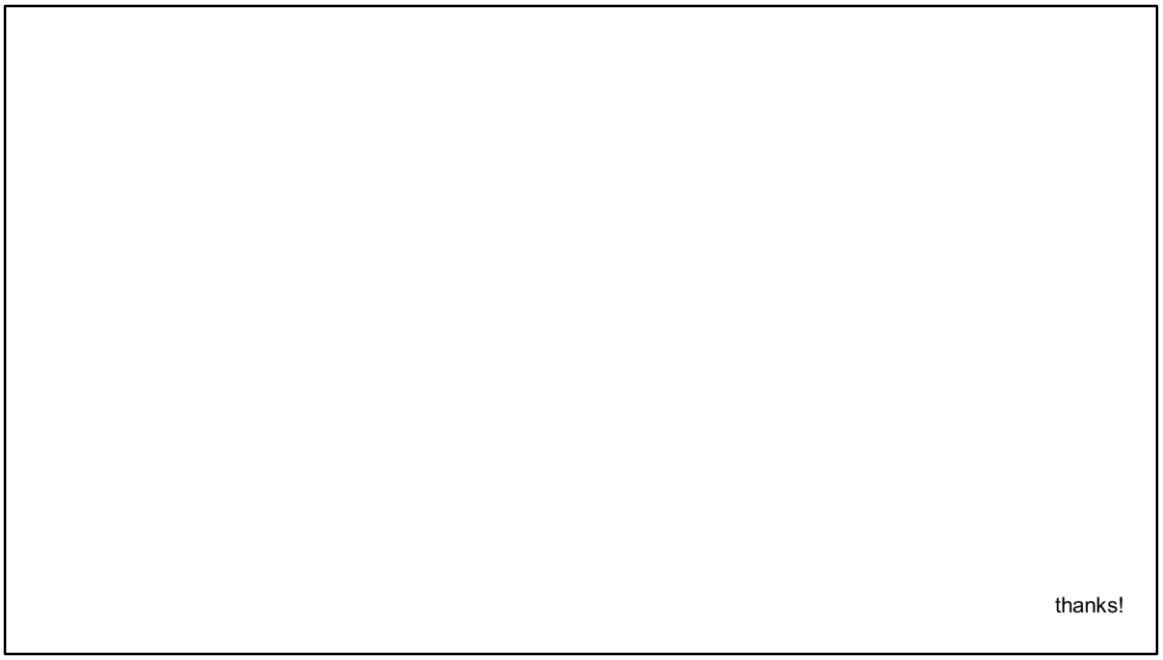
- Trust, but verify
- enable dev, eng, ops to do their jobs quickly but put in checks to ensure security isn't skipped
- Make it okay to push to prod w/ security vulns (within reason) as long as the remaining bugs are placed in the backlog
- If they do skip informally and nicely ask why… maybe your scanning engine sucks
- Most people bypass or avoid security because we're slow and always a step behind

- Build partnerships - security isn't just security's problem
- Must work between silos – don’t expect stovepipes to magically break down
- In fact, the turf wars may get worse as people find religion about tool chain design, tools, and methodologies
- Their motivations haven’t shifted nor has pressures to deliver
- In order for you to succeed you must be empathetic, adaptable, and knowingly respect boundaries others draw between themselves and other teams



- Security needs to be transparent and open
- There really isn't any secret sauce to what we do – we deliver functionality and value to the business
- Share successes and failures with IT peers, let them know you face the same exact issues they do when it comes to delivery and service management

- Here is my request of the devops community
- understand that your security team has a different set of pressures and goals
- Explain what you're doing, why, and how there is mutual benefit
- Be patient with us – we're used to being hit upon the head



Thank you, I really hope this wasn’t a waste of your time