



Hi I'm Jeff – I'm a Founder and the CTO of Contrast Security. And I was the Chair of OWASP for 8 years before I wised up 😊

You may have heard of Continuous Integration and Continuous Delivery?

Well, today I'm going to talk about Continuous Application Security. Here we go...

Title: “Why Your AppSec Experts Are Killing You”

Abstract: Software development has been transformed by practices like Continuous Integration and Continuous Integration, while application security has remained trapped in expert-based waterfall mode. In this talk, Jeff will show you how you can evolve into a “Continuous Application Security” organization that generates assurance automatically across an entire application security portfolio. Jeff will show you how to bootstrap the “sensor-model-dashboard” feedback loop that makes real time, continuous application security possible. He will demonstrate the approach with a new *free* tool called Contrast for Eclipse that brings the power of instrumentation-based application security testing directly into the popular IDE. Check out “[Application Security at DevOps Speed and Portfolio Scale](#)” for some background.

Bio: Jeff Williams is the founder and CTO of [Contrast Security](#), bringing the power of instrumentation and real time analytics to secure your application portfolio. Previously, Jeff was a founder and CEO of [Aspect Security](#). He also served as Global Chairman of the OWASP Foundation where he created many open-source standards, tools, libraries, and guidelines – including the OWASP Top Ten, WebGoat, ESAPI, XSS CheatSheet, ASVS and more. Jeff welcomes hearing from you and may be reached directly at jeff.williams@contrastsecurity.com.

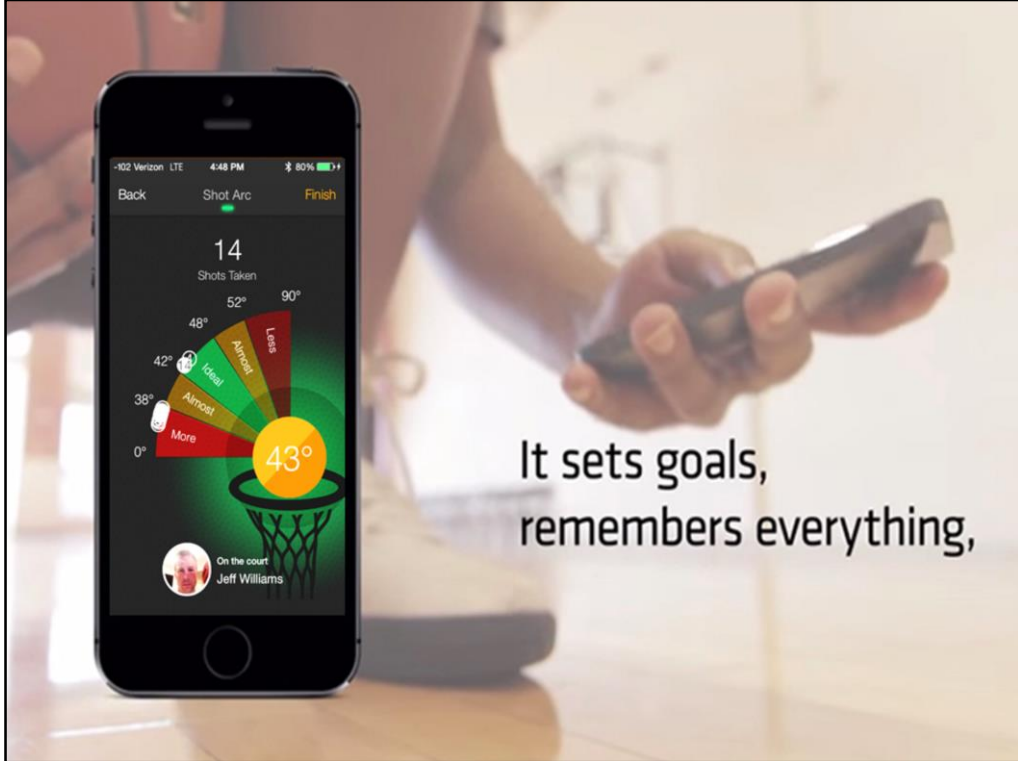


This is not a basketball. This is a basketball coach.

This is Wilson 94Fifty – because the basketball court is 94' x 50'

The 94Fifty has sensors inside it and can monitor your dribble, the arch of your shot, the rotation of your shot, dribbling skills, whether you make or miss.... All the basics.

Connects to your phone via Bluetooth and gives you all kinds of great dashboards.



It sets goals,
remembers everything,

I've been working with it the past few weeks.
It continuously monitors my game WHILE I PLAY – WITHOUT INTERFERING

This one says my shot is too slow and too flat.
I've been playing over 30 years and I did not know that.
And I know it's right because it says right here – THE BALL DON'T LIE

See, 94Fifty has a MODEL of what good basketball looks like.
For example, it knows that the best shooters shoot the ball at a 45° angle with a backspin of 100 revolutions per minute.

AND That's the NEW APPROACH – sensors + model == Incredibly fast feedback loop – far better than you could have with a human coach.

Do you still need a COACH? Sure. But not to watch you while you do shoot thousands of free throws. And not to run drills. Not to track data and do bookkeeping. The coach can focus on making you a strategically better basketball player.

There's no more guessing. Now we are playing “**moneyball**”

Getting to Instant Feedback



https://www.youtube.com/watch?v=4B-HgsT_J_M



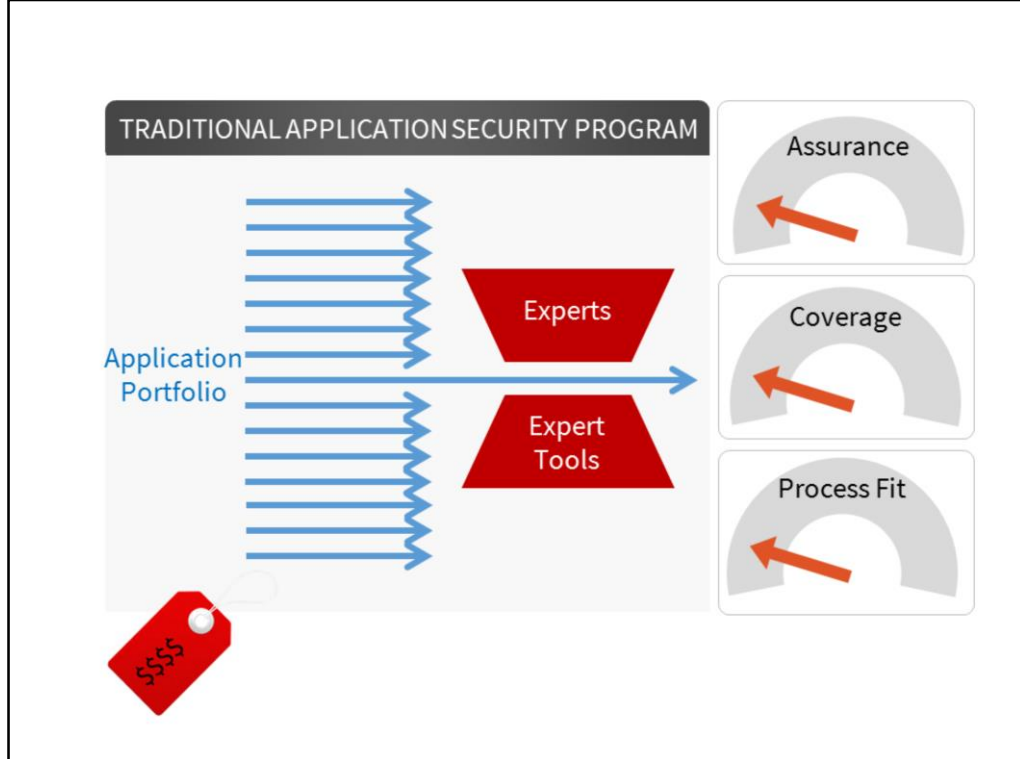
https://www.owasp.org/images/c/c6/2014-04_OWASP_SoCal_-_Continuous1.pptx

I've given two talks on this – bringing moneyball to application security.

The first was at this event one year ago. I gave a talk called AppSec at DevOps Speed and Portfolio Scale. I suggested that appsec specialists are like doctors or coaches, and that we don't have enough of them., and that we can massively improve by leveraging sensors across our development organization.

The second talk was called building an application security sensor network. And it tells the story of how we rolled out sensors, gathered data, and build some fantastic real time dashboards using Puppet, rsync, and a bunch of simple tools – Zap, CYH, etc...

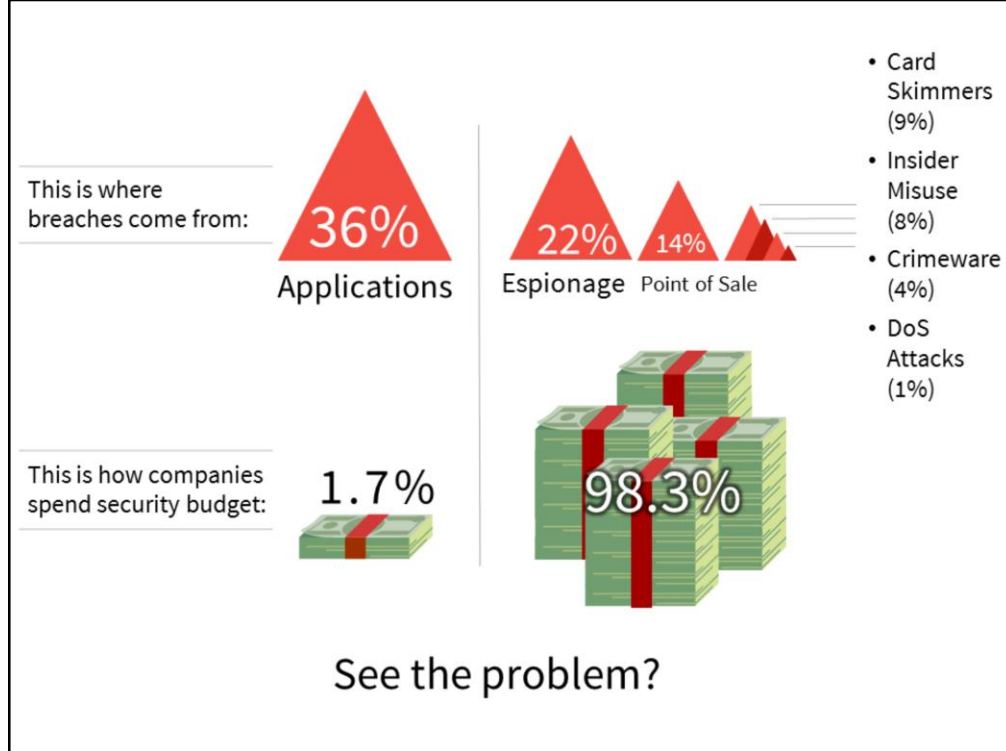
Let's take a second to make sure everyone understands why appsec is broken!



Traditional appsec programs are built around a team of experts and tools for experts. These SO-CALLED “mature” programs have some serious problems...

1. These programs aren't producing any ASSURANCE. At best they deliver a big pile of vulnerabilities – that doesn't give me much confidence.
2. They don't get good coverage over the PORTFOLIO. They tap out at about 10% of the application portfolio.
3. They don't get good coverage over the VULNERABILITIES. Particularly if you're relying on scanners – then you're going to miss almost everything important.
4. But worst of all, they are PISSING everybody off. They're insulting to developers, fear-provoking to executives, and boring for security practitioners.

WHY? They are a bottleneck and a burden. So the natural reaction is to DO IT ONCE. AT THE END. And try to do as little as possible.



Verizon says 36% of breaches are due to vulnerable applications. That's the biggest group of breaches BY FAR.

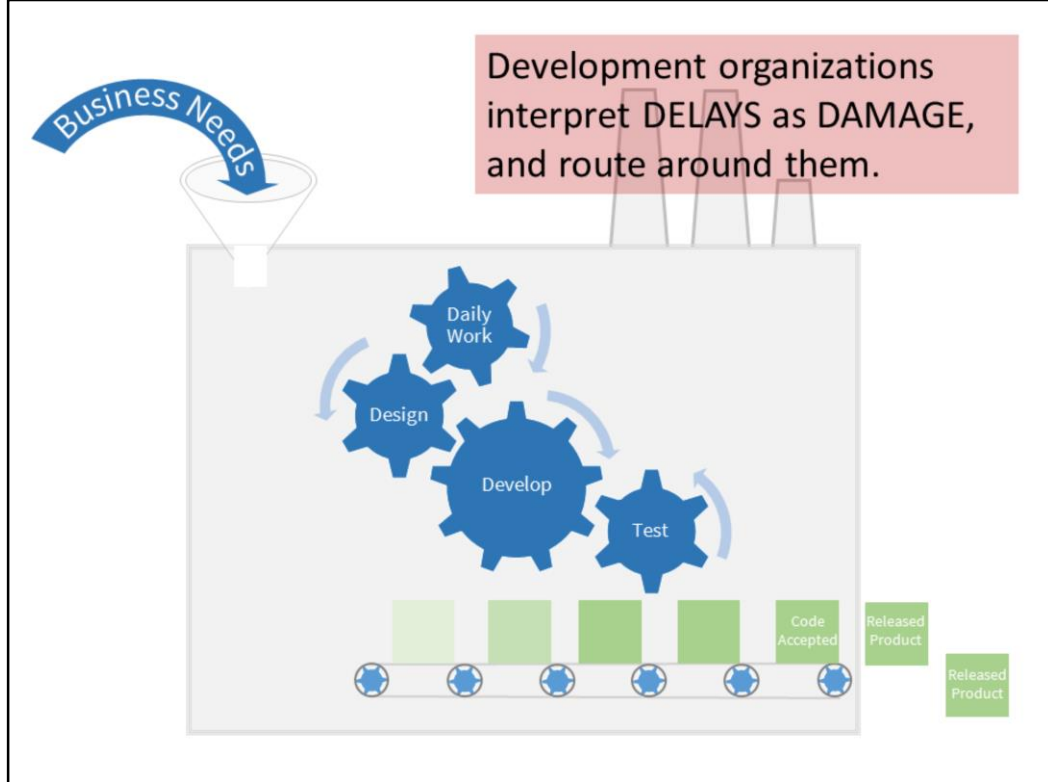
You hear about APT, POS, Skimmers, Insiders, etc... Even DOS more than APPSEC.

But the analysts say only 1.7% of security budget spent on appsec

No surprise: Target, Home Depot, Apple, Healthcare.gov – 3 of the biggest breaches in history have been in the last year alone.

<http://www.informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/>

We have to think different



Waterfall-thinking pervades just about everything in traditional appsec. I want you to stop thinking about a linear SDLC.

Instead, I want you to think about your development organization as a factory that runs continuously – turning business needs into operating software.

Software factories come in ALL SHAPES AND SIZES. Small, large, agile, tools-focused, process-focused, etc...

There are a LOT of different ways to produce secure code.

It's not important that they are all alike. Measuring them against each other is silly. What's important is the quality of their output.

How can we AUGMENT these factories so that we get GREAT SOFTWARE *AND* SECURITY

* The naïve approach is to put in a bunch of gates and security checks

* BUT WE absolutely cannot slow it down, screw it up, or irritate everyone

* Development organizations view DELAYS as damage, and ROUTE AROUND IT – quickest way to be irrelevant

There's a better way...

* Just like the 94-Fifty – we're going to focus on the sensors and the model.



We are not so unique. Many industries have gone through a revolution where production outstripped quality. Think auto industry in the 70's.

But they caught up. How? It starts with SENSORS

They instrument EVERYTHING.

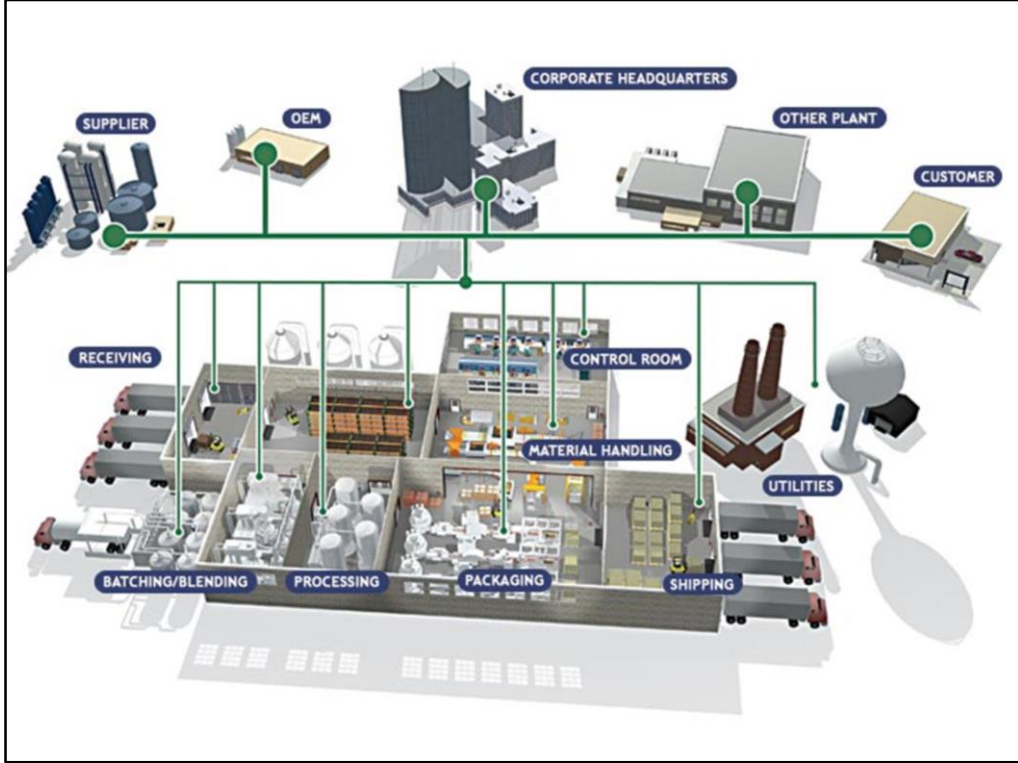
These are sensors for all sorts of things... Vibration, sound, moisture, temperature, cameras, etc...

They can tell that a bearing is going to break way before it actually does.



Now imagine all the connectors and ADAPTERS that they have to have to do all this

They've got all kinds of signals and data to process
So a large chunk of the problem is getting the sensor data standardized



Now all those sensors are connected to Programmable Logic Controller (PLC) or Programmable Automation Controller (PAC)
They gather up all the raw data and they process it into standard forms.

Like the 94-50 they have a MODEL to compare against.

They do realtime ANALYTICS to check for problems with the equipment AND the product.



And they pull all this data together to create awesome dashboards and alerts

[[ALLOWS you to maintain quality at high speed]]

They have completed a realtime continuous feedback loop.

Anytime anything goes wrong in the factory, they get immediate alerts and can fix the problem.

[[THAT'S the pattern – sensors, adapters, analytics]]

Is it possible to cobble together
noisy, spotty, difficult, expert-only sensors
and get good results?

No.

No -- You have to **start with GREAT SENSORS!**

For many years I was a sensor – a MANUAL code review sensor
SAST and DAST are sensors – they are a part of your software development organization.
But they are noisy and awful

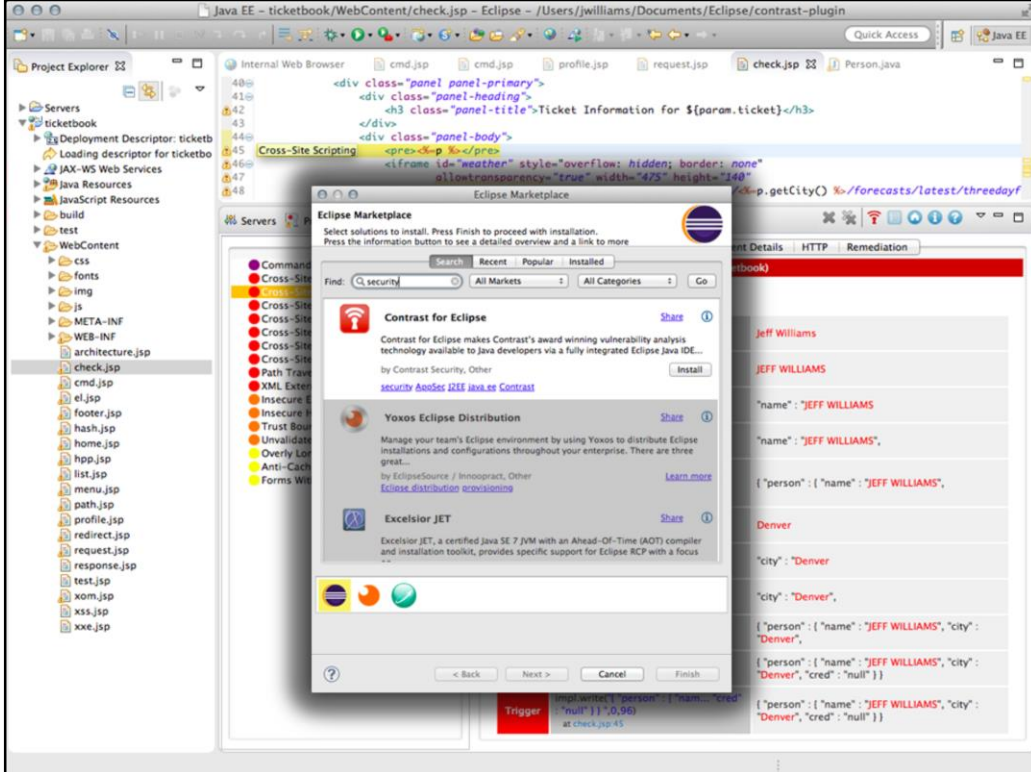
Accuracy matters! I've heard a LOT of people arguing that if you have crummy sensors, you might try running a lot of them in place, correlate them, and prioritize the ones that get multiple hits.

* First of all it is tons of work – every tool you add has onboarding, tailoring, execution, merging, and triage.
running multiple tools takes HUGE amounts of time

* Second you are going to lose the things that each tool does uniquely well!

What you need is SIMPLE accurate sensors...

TO THAT END... I am incredibly excited to be able to announce a new FREE application security sensor unlike anything you've ever seen.



This is Contrast for Eclipse – or CFE

(BTW – I verified with the OWASP Board that it’s absolutely okay to talk about this at OWASP – a lot like Burp FREE actually).

This is NOT SAST or DAST – it’s software instrumentation – radically different
GARTNER calls IAST a “breakthrough” technology and one of the 10 critical security technologies of 2014.

I’m going to explain while I show you...

MAKE EVERYTHING ABOUT FAST – FAST IS what changes everything (FUN)

1. Fast to install – no config
2. Fast to feedback
3. Fast to retest
4. Super accurate because it has more information – code, HTTP + runtime data flow, libraries, config, backend

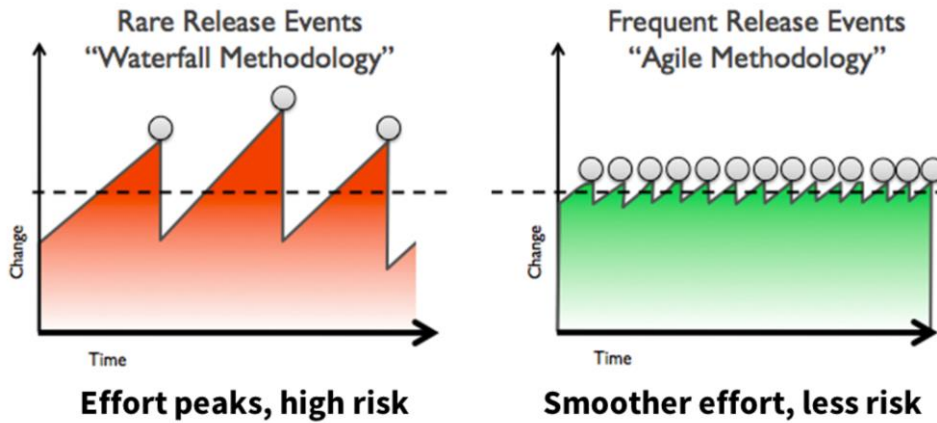
1. NOT a static or dynamic tool. This is a new technology called IAST.
2. Works in REALTIME – hundreds of times faster than both dynamic and static
3. Works on HUGE applications
4. Analyzes ENTIRE APPLICATION – including libraries, frameworks, and runtime
5. Non-security experts can use this today!
6. Stop worrying about tracking vulns. It’s super easy to find them again.

Continuous...

“To survive this...

- We have to monitor
- We have to report
- We have to adapt, in realtime, to our mistakes.”

- Dan Kaminsky



You've heard of Continuous Integration and Continuous Delivery? WHAT is continuous all about -- Well, it's like BALANCING ON A BIKE – lots of small corrections – much more efficient. The good news is that this is already probably happening in your organization.

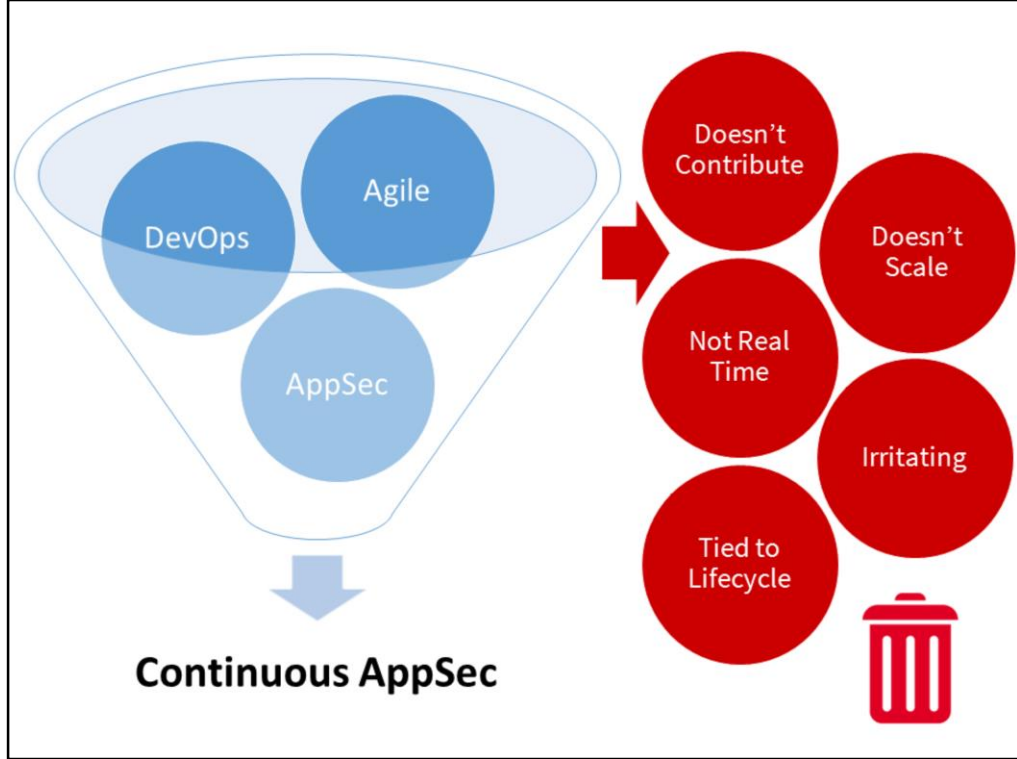
Ok, so that's a great developer tool. But it's just one sensor. You can't just drop a single sensor into a factory and expect everything to be fixed. You need to monitor all aspects of software production. The CULTURE of instrumentation has catch FIRE.

My knee jerk reaction was that continuous processes were dangerous for security. But the more I learn, the more I realize that they are critical for security.

Nick Galbreath from Twitter and I had a great conversation about this and he said CONTINUOUS practices force software projects to put in the processes and structure to automatically test the quality of their software. In effect, they create the infrastructure to do security well.

Now I'm all in – just imagine if we could, AT ANY TIME, provide a clear, supported, defensible assurance argument.

(Companies like New Relic and AppDynamics are using instrumentation for performance engineering. You used to need an expert with expert tools.... And now you can do it for yourself – everywhere in the development process.)



My colleagues at Aspect Security have spent the last year working with clients to help them build CONTINUOUS APPSEC programs.

Basically they took....

- 1) Software development trends like Agile and DevOps
- 2) Security methodologies from the Orange Book, maturity models from the 90's, C&A, OWASP, and others.

They chucked anything that doesn't:

- Directly contribute DIRECTLY to "security"
- Doesn't scale
- Doesn't work in real time and continuously
- Tied to lifecycle stages or waterfall

For example – security requirements, security architecture docs, security test plans???

And we refactored the rest. We came up with....

Continuous Application Security...

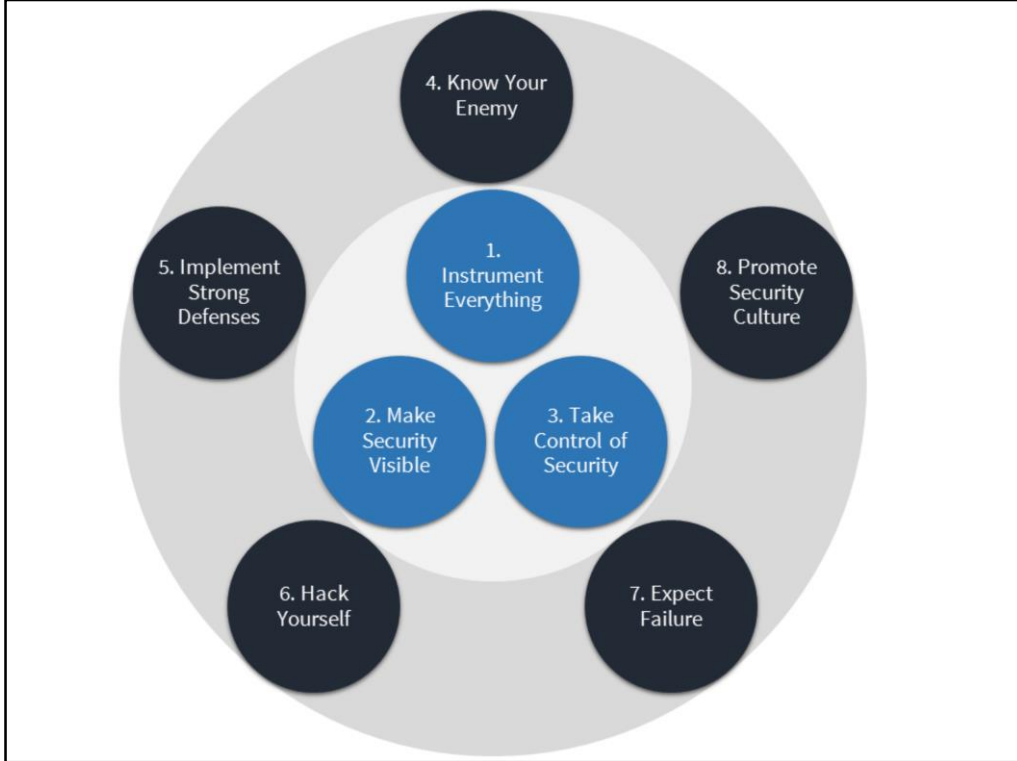
- Eight activities that *enhance* modern software development to produce “secure” code
- Rely on *automated sensors* to gain continuous and real time feedback and visibility
- Performed *continuously* throughout the development organization.
- Focuses on whether the security you want been *achieved*, not whether you do what others do
- No *experts or “gates”* in critical path of software development.



THEY refactored the rest into eight activities we call CONTINUOUS APPLICATION SECURITY

Here's the concept...

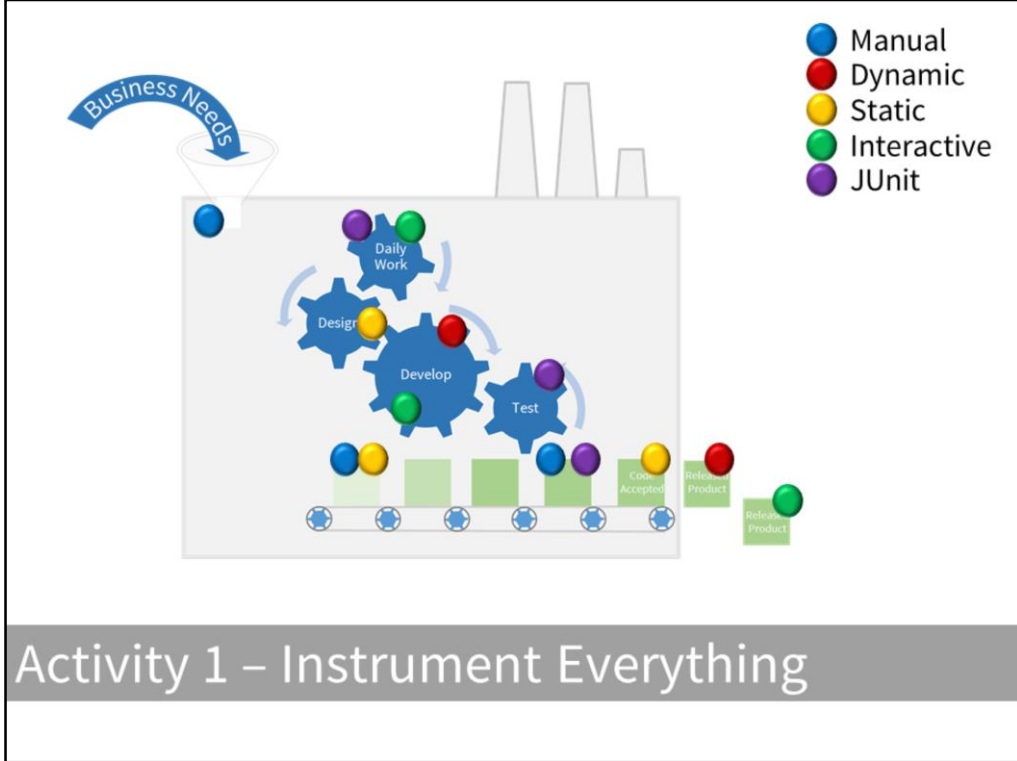
- ENHANCE the development organization – these activities can be added to any software process.
- AUTOMATED sensors for continuous real time feedback throughout the organization
- MEASURE whether you achieve security, not whether you do what others do
- COMPATIBLE – no experts or gates. Primary responsibility on developers.



These are the 8 activities, but today we will focus on just the core 3

[[THE FASTER you iterate the core practices here – model, sensors, analytics – the faster you get secure]]

The CORE: Create a tight feedback loop – sensors, model, dashboard – JUST LIKE GOOD OLD 94Fifty. CONTINUOUSLY checking in real time whether you are doing what you set out to do. In the end, this is probably enough, as most of the other stuff will happen naturally in most organizations.



INSTRUMENT EVERYTHING

I SAID – IT STARTS WITH GREAT SENSORS. I talked a lot about this last year...

INTRODUCE IAST is the future – FRAMESAST AND DAST as legacy

Remember the actual industrial factory sensors? It's easy to make sensors to measure certain aspects of security. You can wrap existing tools like Zap, Burp, CYH, Dependency Check Junit, etc...

But you should use sensors to measure exactly what you care about – THE MODEL. Not just running a tool blindly.

Whatever questions you have about security – answer them with a sensor continuously.

To make it a sensor, you have to do two things:

- 1) extract the data and put it into a common form. Remember those adapters? Well we need them.
- 2) You also need to get the sensor data and send it to a centralized location – an Application Security Data Warehouse.

What they try to sell us:



Super smart pizza boxes ^[1]

What we would buy:



Software sensors with centralized intelligence ^[2]

-- Alex Stamos, Yahoo CISO



This isn't just for the unicorns – Google, Etsy, etc... Anyone can do this.



Activity 2 – Make Security Visible

2: Make Security Visible

Sensors are great. And if you're monitoring one thing, then a great sensor is all you need.

You know the OWASP mission? Know why it's on visibility – transparency? Fundamentally appsec is a problem of asymmetric information. It's a market failure. And creating visibility is how to fix that market.

Let me tell you a story about a large financial organization which has adopted CAS. When we started, their model wasn't well defined, they did triannual reviews of applications (TRIANNUAL!!), and their only dashboard was a total count of vulnerabilities.

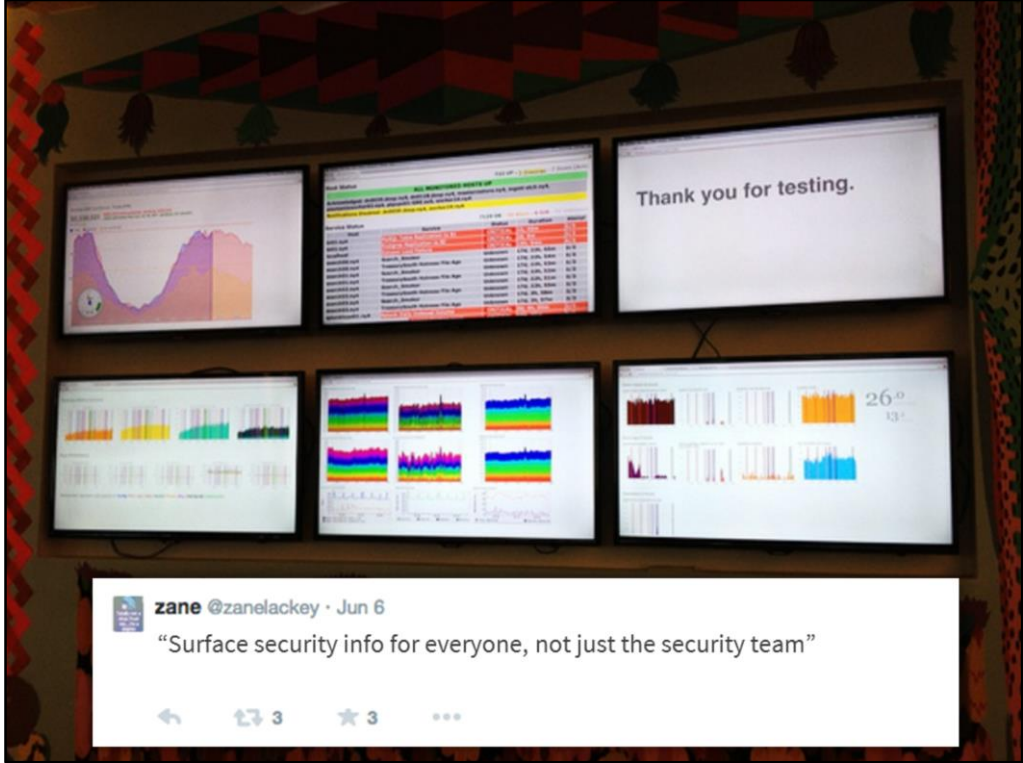
The Aspect Team has helped them integrate powerful sensors directly into their build infrastructure to verify some of their model continuously. Some sensors are commercial, some open source, and some custom. We created a custom dashboard in SONAR that makes security visible to everyone. AND – THEY just enabled security to “FAIL THE BUILD” for better feedback.

They are on a great path and it is already showing results. They are expanding coverage of apps and vulns without increasing budget.



Yelp handles CSP at SCALE

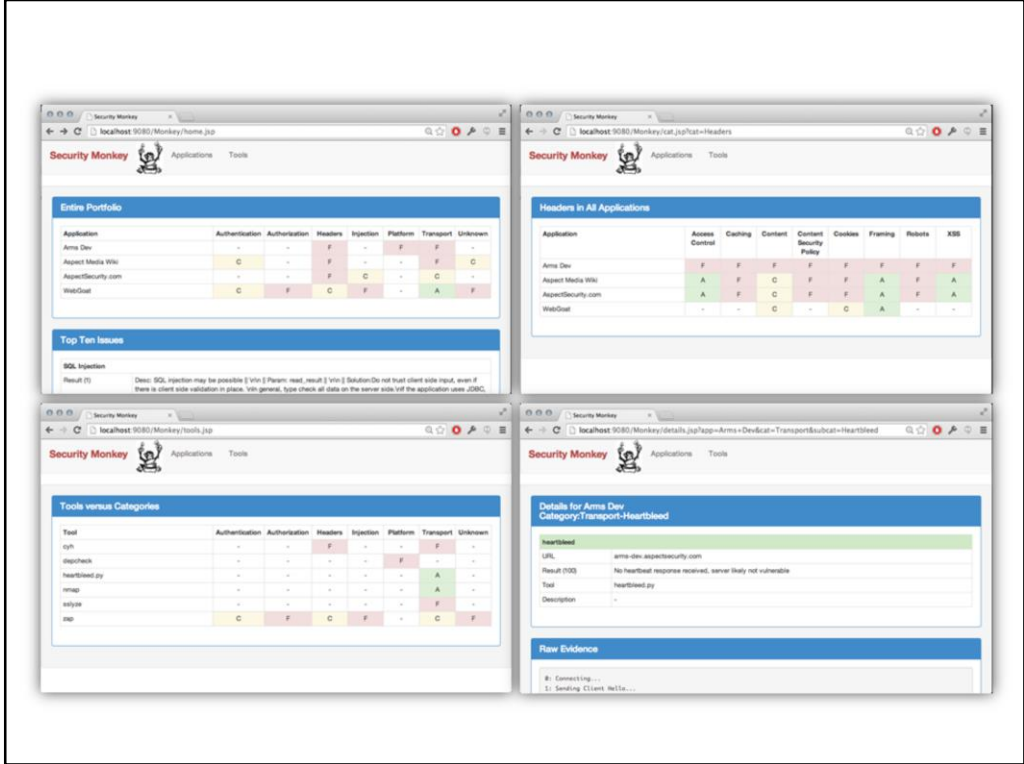
1000's of CSP messages per minute.



This is the security dashboard at ETSY

They've surfaced appsec to everyone, not just the security team.

They all work together to make sure the metrics are right.

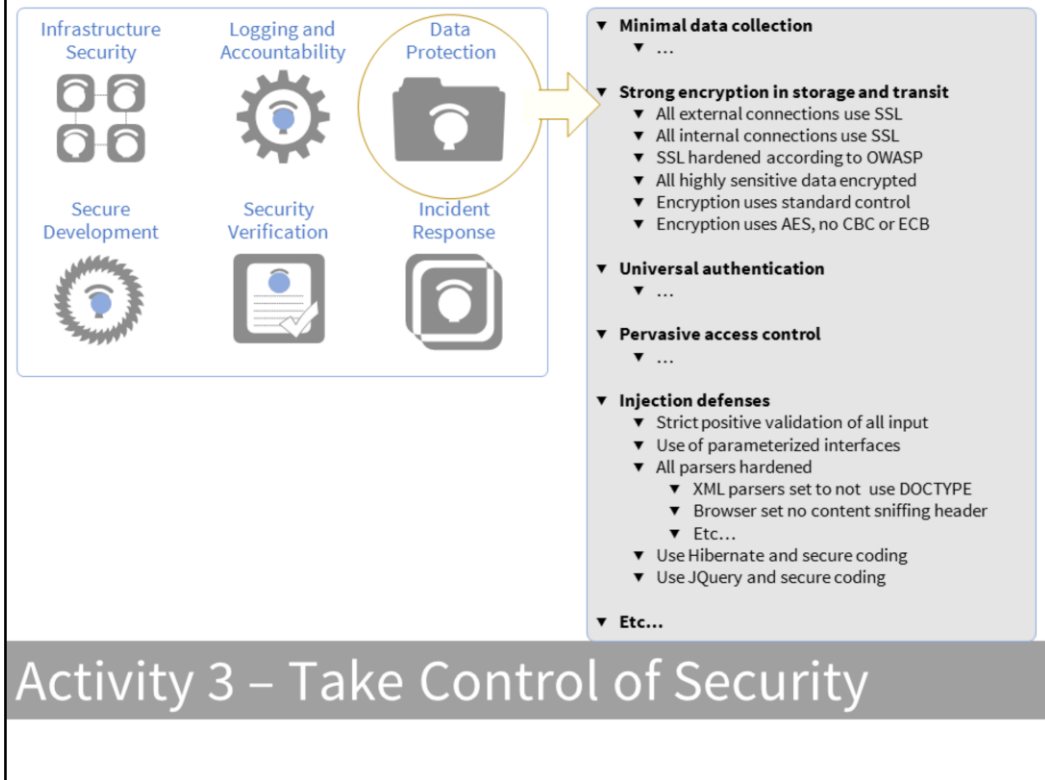


This is not as hard as you think... I talked about this in LA

This sensor network relies on Puppet to manage sensors across a wide variety of platforms. Rsync back to a centralized data hub. And a set of “digesters” to translate the tool output into a common appsec format.

I hooked up OWASP projects like ZAP and Dependency Check. I hooked up FindBugs and PMD. I even hooked in heartbleed checkers, CheckYourHeaders, and others.

All these tools are reporting in real time across a portfolio of applications. Constantly up to date.



Activity 3 – Take Control of Security

TAKE CONTROL

You're in control when YOU define what security means.

THE security folks at FRS have done a nice job of working this aspect of security.

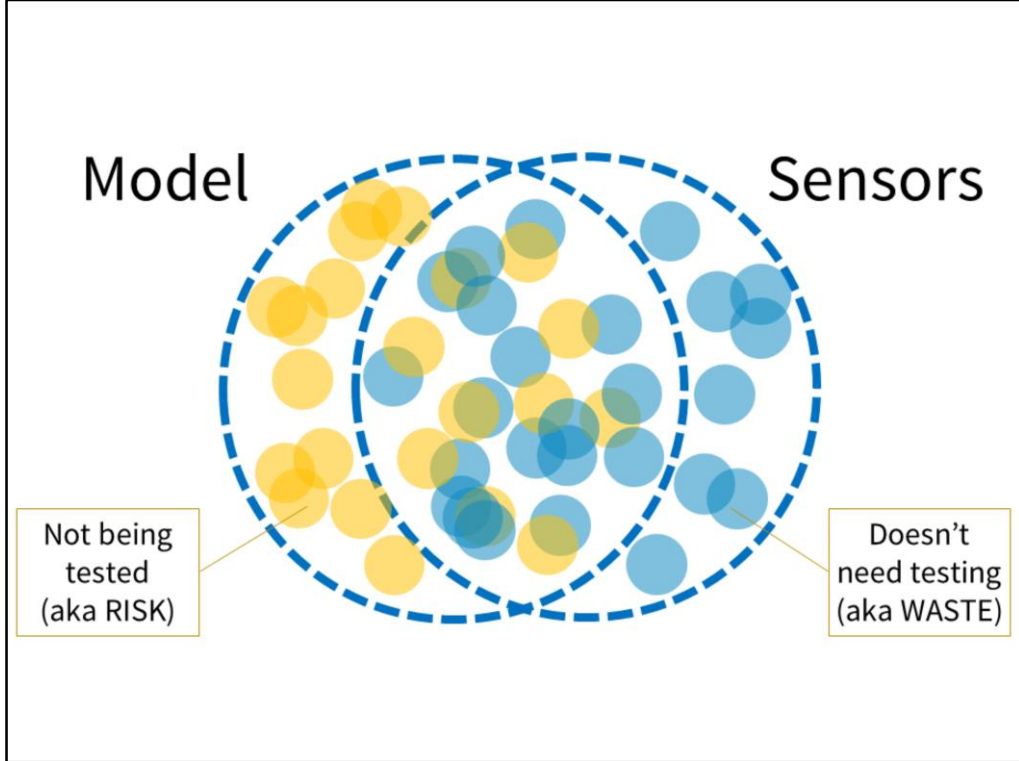
So many organizations just rely on an external standard like PCI, or a tool, or some dumb thing like the OWASP Top Ten. In fact, they have absolutely no idea what security means.

A lot of people think you should START with the model. But it doesn't really work.

You have to start with SENSORS – get some real data, and then back into a model.

You're going to have to define and improve your expected security model. Put something in place and evolve it.

Here we see....



In CAS, we call this your EXPECTED SECURITY MODEL.

It could be a list, a spreadsheet, or something more sophisticated, like a story or an assurance argument.

Most organizations just test, so they don't know that they're missing all the yellow stuff.

And most are also testing for things that they don't really care about. The blue stuff.

And at the end, you end up with a mess.

Capture your current model. What defenses do you have? Why are your defenses there? What do you test for? Evolve your strategies.



That's the CORE. 3 activities to create a tight feedback loop -- model, sensors, dashboard.

I DARE YOU TO TRY IT – give it a try on a single vulnerability. Clickjacking. I don't care. Once you try it you'll never stop.

The next 5 activities enhance that feedback loop.

Mostly these will happen naturally once you get a strong core in place. But it's helpful to know where to go next.



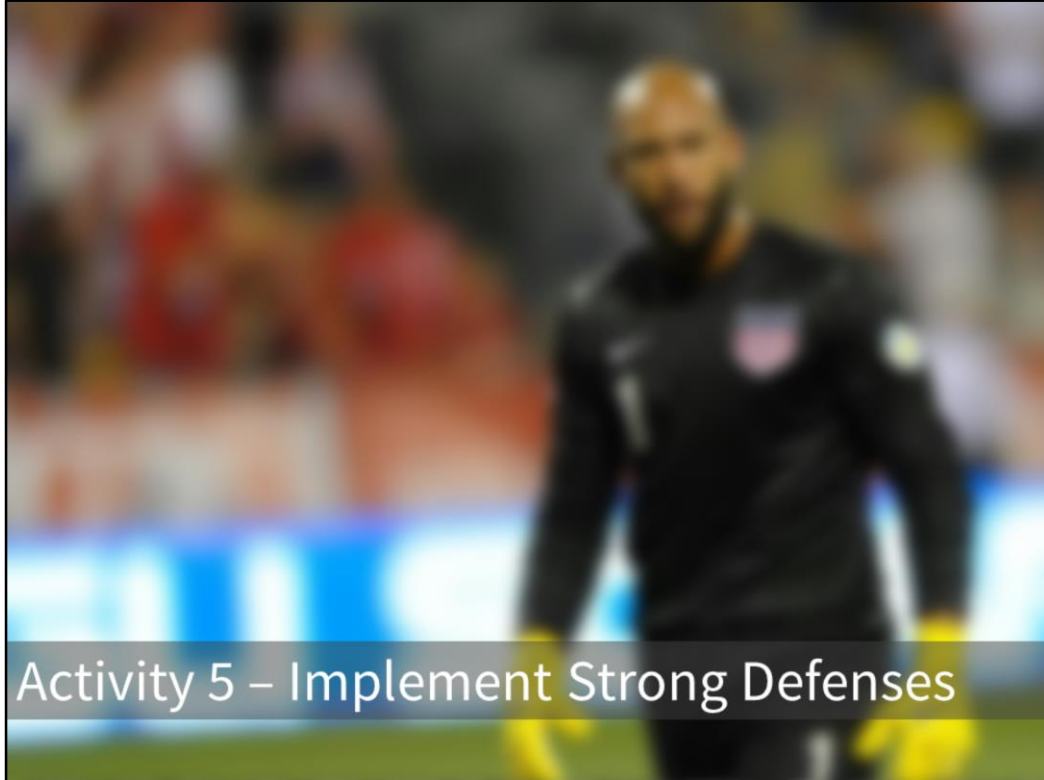
Activity 4 – Know Your Enemy

Know Your Enemy means that you are actively trying to understand the threat to your businesses.

This is the first way to improve your ESM by adding new threats.

You need to monitor good sources of application security threat information and challenge your ESM!

What are some good sources of this data??? (OWASP, BlackHat, FSISAC, Verizon, etc...)




Activity 5 – Implement Strong Defenses

Having a strong set of standard defenses makes everything else in application security easier and better. This allows you to simplify your model AND your sensors

You should establish your own Enterprise Security API and sensors to verify it. POSITIVE!!

What's easier – trying to exploit every possible XSS vector in your enterprise.... Or checking your UI's to see that they use an approved Encoder?

This simplifies your model and makes for better, more accurate, easier sensors.



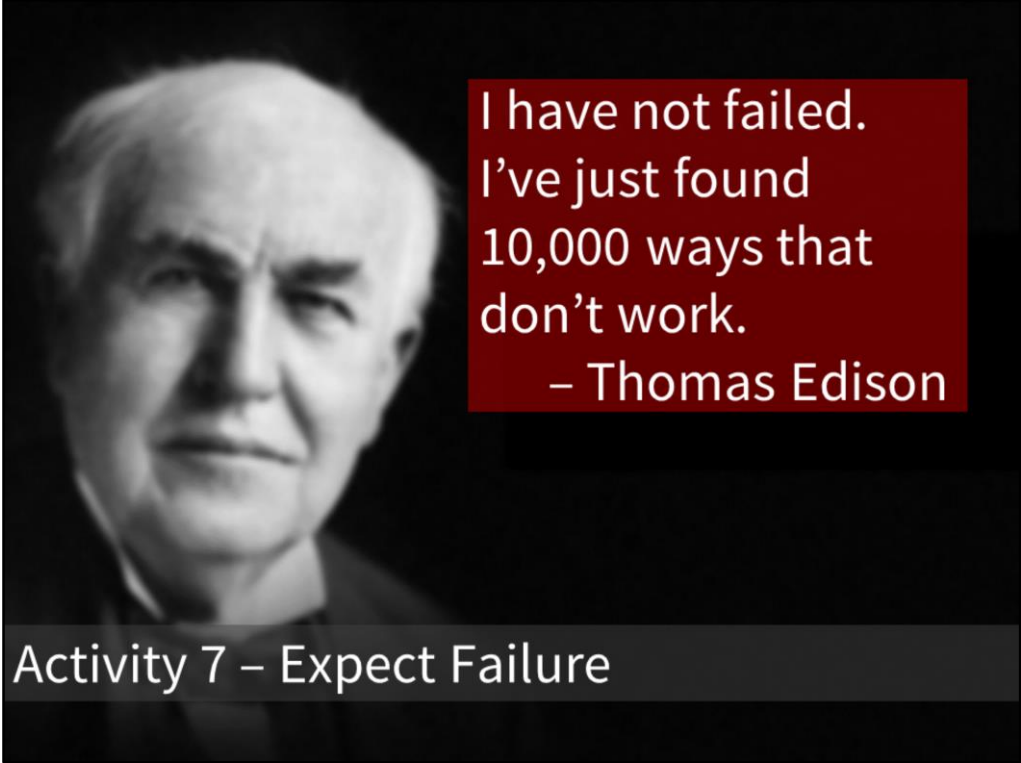
Activity 6 – Hack Yourself

TRANSFORM your penetration testing practice.

- 1) Hack the model
- 2) Verify what you don't have sensors for (new risks)
- 3) Produce sensors (note: Mozilla wants vulns in ZEST)

As you replace penetration tests with sensors, you'll naturally start to look for gaps and holes in two places:

- You expected model
- Your security defense implementations




Expect failure means that you have prepared to be exploited.

You are ready not only to detect attacks, but also to respond.

Really this is just part of your ESM – detection and response are part of your security strategy.

Hopefully you'll stop them, but if someone successfully attacks you:

- 1) You'll have a better chance to detect the attack
- 2) You'll have the data to respond intelligently and quickly (I'm talking to you Home Depot)
- 3) You'll be ready to communicate to the media

A photograph of Peter Drucker, an elderly man with glasses, wearing a dark blue suit jacket over a white shirt and a dark tie. He has his hands clasped in front of him and is looking slightly to the right of the camera with a thoughtful expression. The background is a plain, light-colored wall.

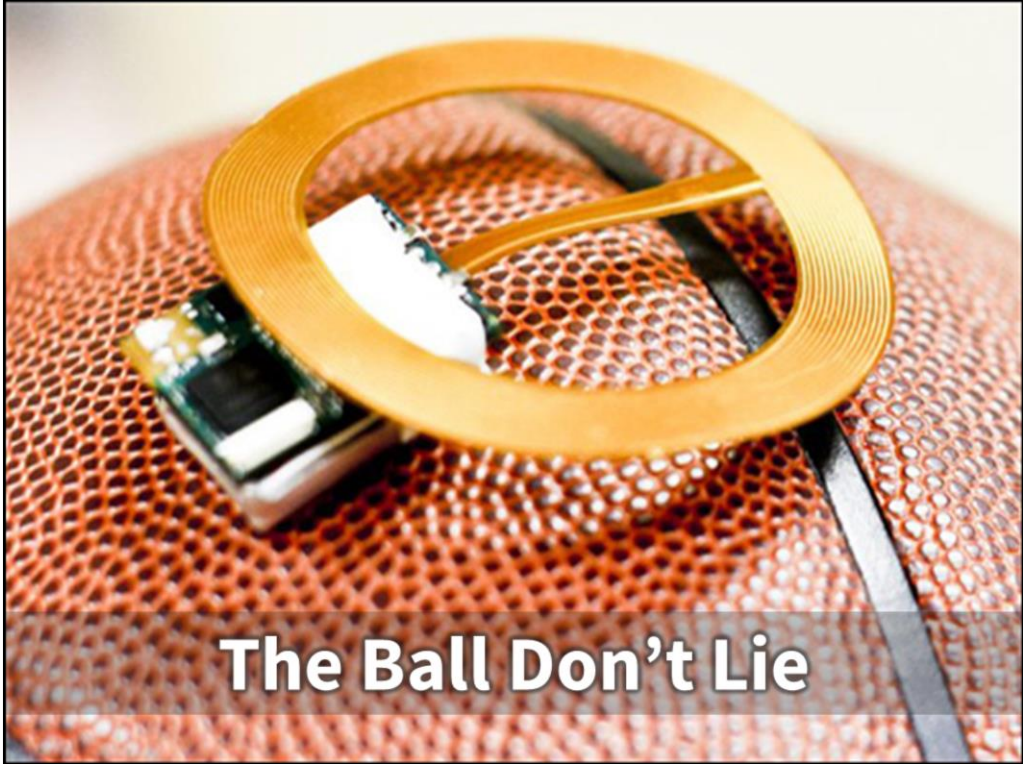
Culture eats
strategy for
breakfast.
– Peter Drucker

Activity 8 – Promote Security Culture

If you've done the other 7 practices, then 8 is going to come pretty naturally.

But you can accelerate that security culture with appsec training/eLearning

You'll also want some support from executives.



The Ball Don't Lie

When you have Sensor-> Model -> Analytics – really humming – that’s where security comes from.

You don’t have to wait for weeks to learn. You get immediate feedback. You can make adjustments in real time.

You don’t have to go back and watch the tape.

AppSec experts need to be coaches and toolsmiths.

It’s a new world. All the data you need is right there, waiting for you to gather it up and put it to use.

Today, WE HAVE GIVEN you a very powerful sensor and a roadmap for where to go with it

Please contact me if you’d like to help me build more CAS materials.

Thank you!

I am easy to find and I am here to meet you

Jeff.williams@contrastsecurity.com

@planetlevel



Don't forget to pick up your
free handbook and free Eclipse plugin.

