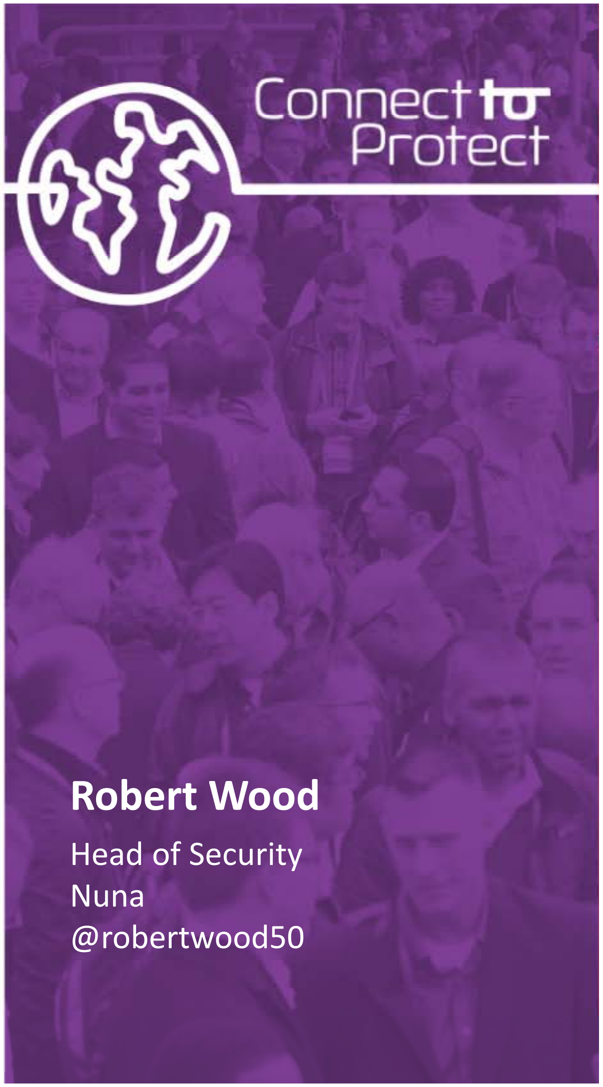
RSAConference2016

San Francisco | February 29 – March 4 | Moscone Center

SESSION ID: CSV-R02

Security Program Development for the Hipster Company



Connect to Protect

Robert Wood

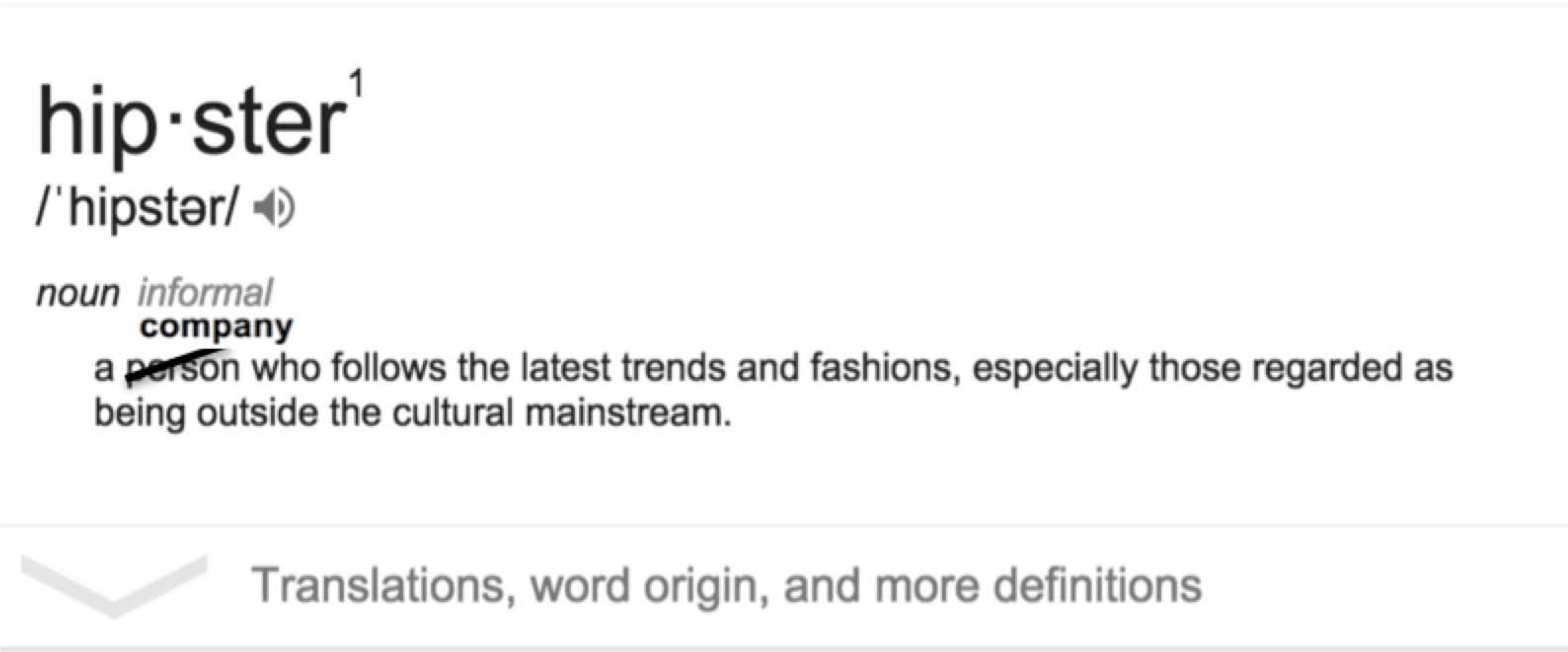
Head of Security
Nuna
@robertwood50

#RSAC

whoami

- Head of Security at Nuna Health
- Former Principal Consultant at Cigital
- Background focused in red teaming/alternative analysis

Our Focus Today



hip·ster^{1}

/ˈhipstər/

noun *informal*

a ~~person~~ company who follows the latest trends and fashions, especially those regarded as being outside the cultural mainstream.

Translations, word origin, and more definitions

Example Company Setting

- AcmeBill is building a hip new financial portfolio management/monitoring tool on AWS, subject to PCI compliance
- Using Google Apps, Slack, Todoist and Box to run the rest of the business
- Compliance and security relevant data
  - Primarily in AWS with the product
  - Sometimes employees need to share, use, chat, display sensitive data using Google and Slack
- You’re in charge of managing the risk around data and it’s potential unauthorized use or disclosure

Assumptions

- DevOps (or similar) culture
  - Fail fast and fail small
  - Heavy reliance on automation
  - Openness and collaborative culture
- Early stages of security program development

Agenda Today

- Who should be thinking about these issues?
- Compliance considerations
- Security considerations
- Practical steps for getting started

Who Should Care

- Executive team
- Engineering lead(s)
- Security/compliance team(s)

RSAConference2016

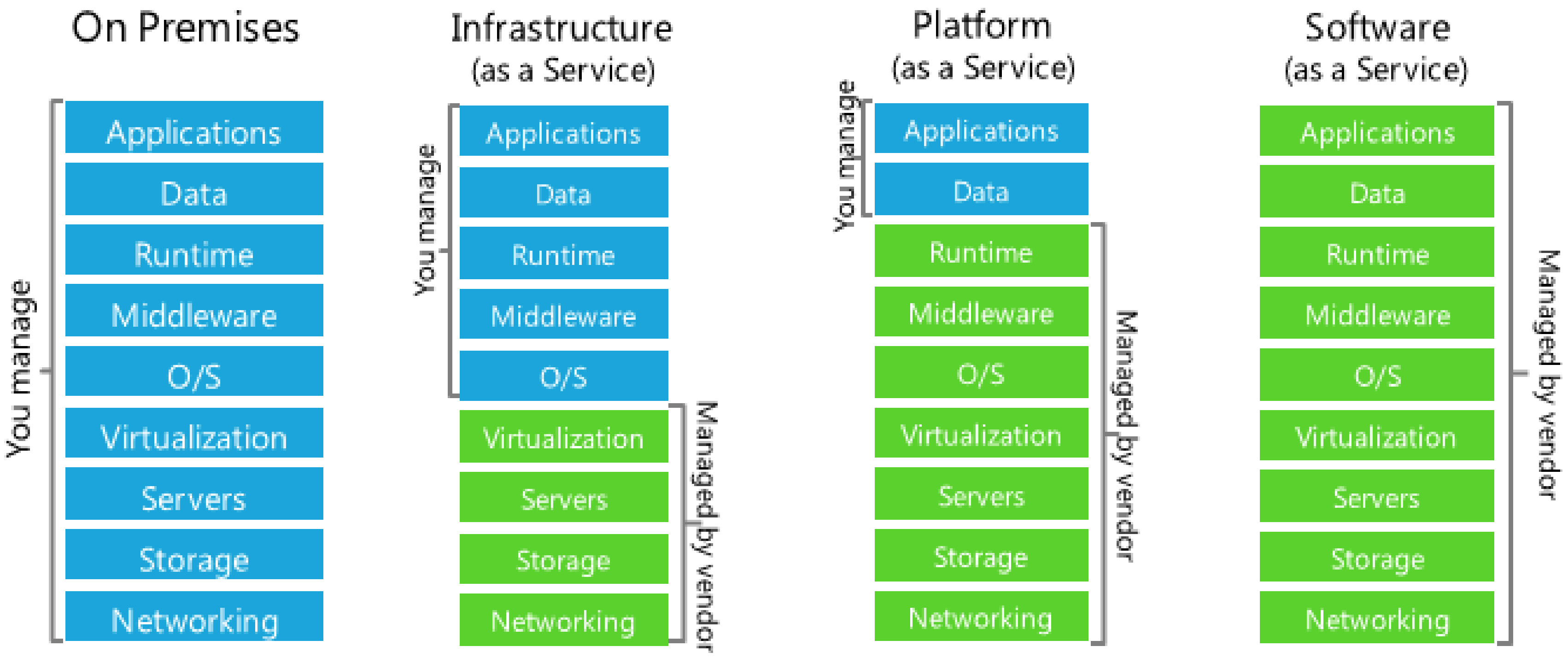
#RSAC

Tackling Cloud Compliance

Why/When Does This Matter?

- Need to consider these factors when:
  - A product you're building will be built on top of cloud services
  - A cloud service you're using will handle data that is subject to compliance requirements
- Risk ends up being shared between you and your provider
- More of a partnership than a transactional purchase

Cloud Ownership Model



On Premises

You manage
- Applications
- Data
- Runtime
- Middleware
- O/S
- Virtualization
- Servers
- Storage
- Networking

Infrastructure
(as a Service)

You manage
- Applications
- Data
- Runtime
- Middleware
- O/S

Managed by vendor
- Virtualization
- Servers
- Storage
- Networking

Platform
(as a Service)

You manage
- Applications
- Data

Managed by vendor
- Runtime
- Middleware
- O/S
- Virtualization
- Servers
- Storage
- Networking

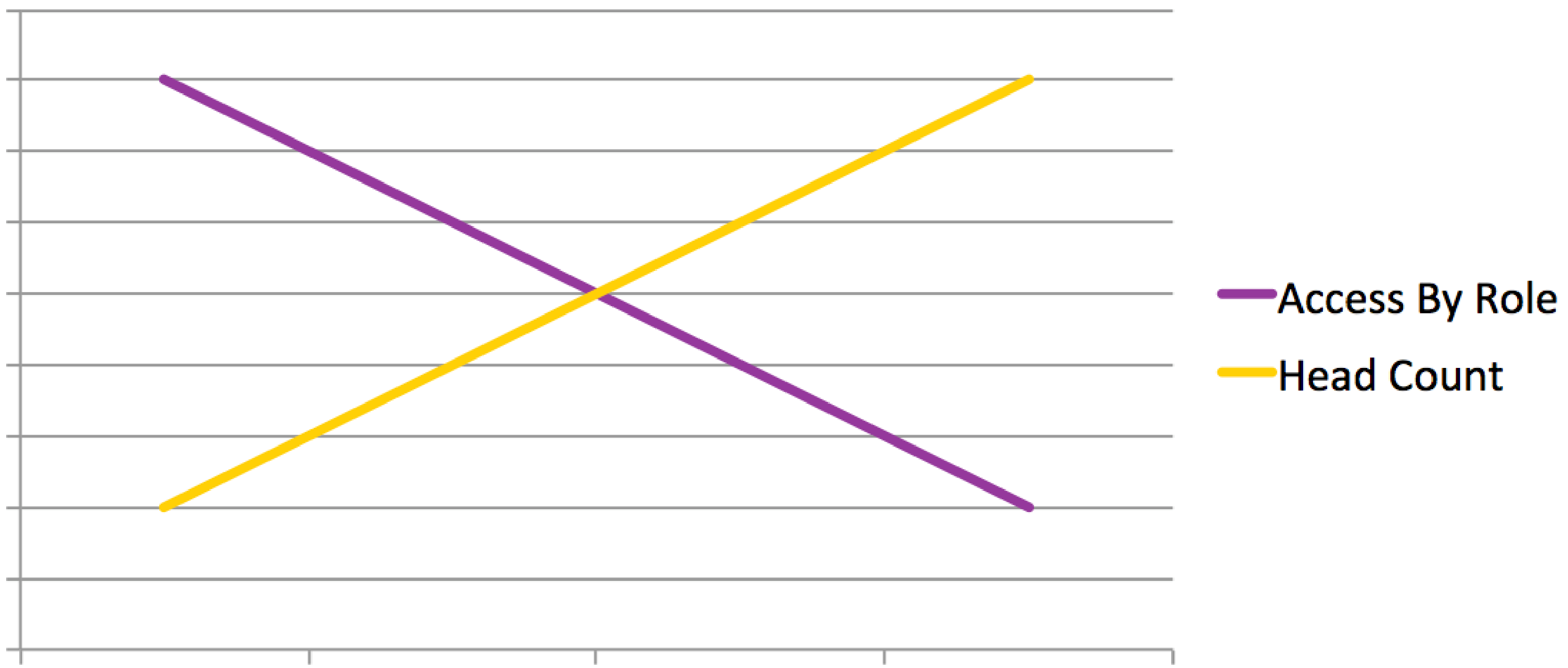
Software
(as a Service)

Managed by vendor
- Applications
- Data
- Runtime
- Middleware
- O/S
- Virtualization
- Servers
- Storage
- Networking

NUNA

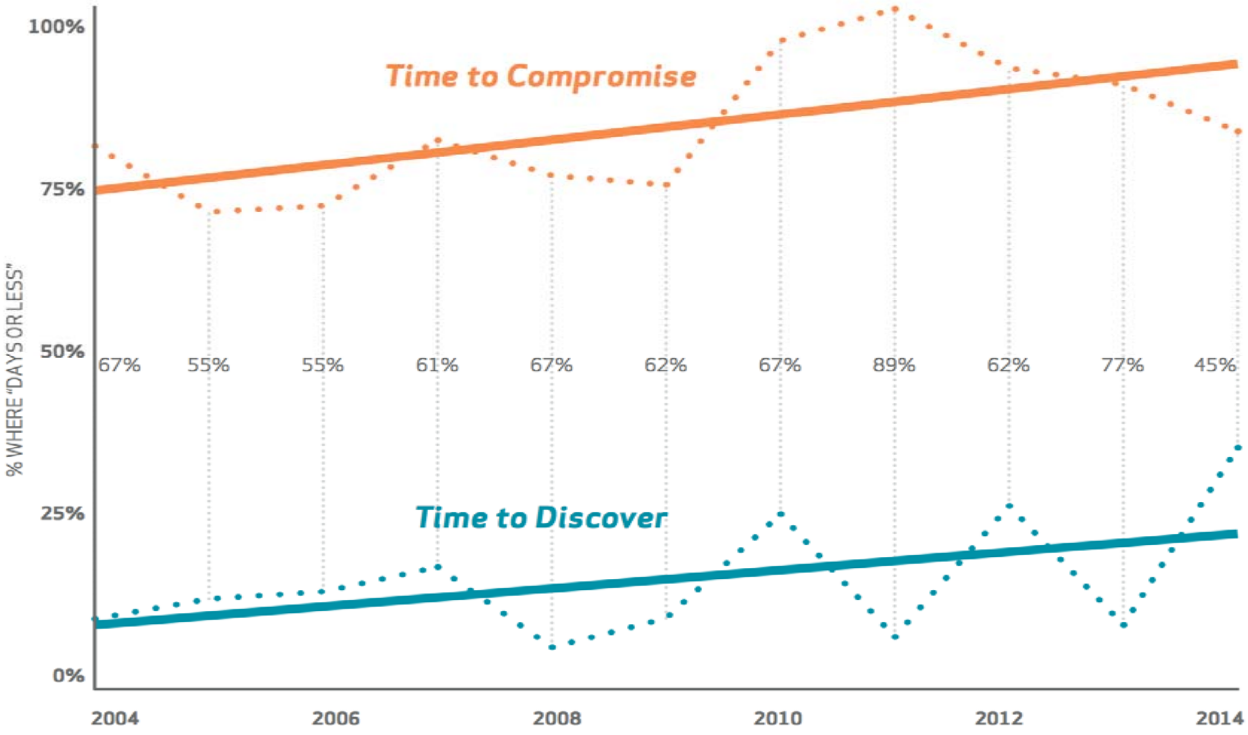
Pivoting to Security

Increased Access

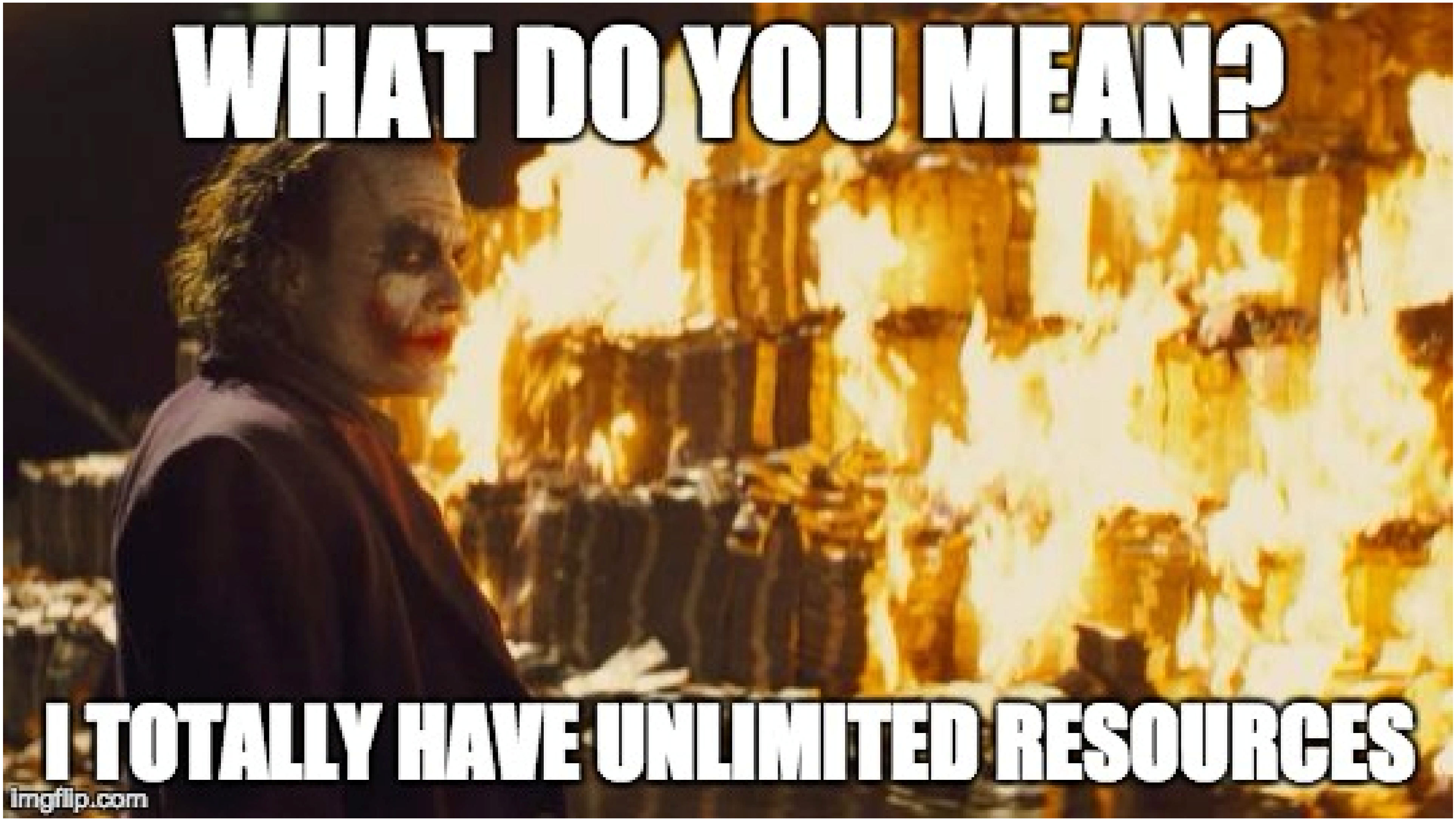


Blind Monitoring Fail

Attacker Economics



NUNA


WHAT DO YOU MEAN?
I TOTALLY HAVE UNLIMITED RESOURCES
imgflip.com

What are the Real Goals?

- Not possible to secure everything
- Set and refine OKRs specific to security and compliance:

Objective: manage risk to an acceptable level based on threat profile

Key result: reduce all critical vulnerabilities in Internet-facing services from 10X per 10kloc to 2X per 10kloc

Objective: level up visibility with centralized logging

Key result: all product and associated infrastructure logs should be captured by centralized log manager along with operational cloud services.

Security Touchpoints

Security Touchpoints

- Heavy weight (infrequent) touch points:
  - Security requirements
  - Static analysis
  - Penetration testing
  - Full blown red teaming
  - Incident response
- We can break each of these into leaner approaches
- Each of these still have a place in bigger picture, but can’t be done as frequently

Security Requirements (Short Term)

- Feature-by-feature security requirements (including abuse and misuse cases) can be time consuming to generate and hard to track
- Instead generate a set of specifications that:
  - Identify what the system should NEVER allow
- Engineers can review functional software requirements against this list to determine compatibility

Security Requirements (Long Term)

- Look to consolidate development environment and build security controls at the framework/library level
- The goal is to make the adoption of security requirements as easy as possible for engineers:
  - Transparent
  - Convention
  - Service layer

Static Analysis

- Measure and set goals around code coverage
- Improve over time with security focused linters within CI/CD:
  - Insecure API usage
  - Insecure crypto usage
  - Vulnerable dependencies

Penetration Testing

- Shift away from a standalone test with written report
- Shift towards continuous models
  - Bug bounties
  - Vulnerability scanners run constantly with developer-centric feedback loops (e.g. JIRA tickets)
  - Translate true positives into CI/CD regression tests
  - Train QA to utilize subset of security tools and interpret the results

Red and Blue Teaming

- Responding to major attacks (data breaches) needs to be an organization-wide effort
- Simulate susceptibility and response efforts through collaborative table tops:
  - Security team can profile adversaries and introduce doomsday scenarios down to everyday security issues
  - Get representatives from many teams involved
  - Increases awareness and highlights cross-team thought processes
  - Leverage previous incidents and associated root causes to help drive potential focus areas

Let’s Start Building

Three Measurement Axes

- Depth
  - Quick scan vs. really deep analysis (manual or otherwise) on a single thing?
- Breadth
  - How many parts of the business (how many apps, IP’s, etc. are covered)?
- Knowledge share
  - How many people know how to do this?
  - How many people receive the results?

Managing Risk

- Spending on a security activity should have an impact on risk:
  - Protect
  - Detect
  - Respond
  - Recover
  - Transfer

AcmeBill Applied

- Compliance relevant data stored in AWS, Google Apps and Slack
  - Are those vendors PCI compliant for what they control?
- Translate lean security touch points through DevOps practices to build the product on AWS
- Collect logs from all services to ensure that PCI data isn't being used outside of defined boundaries
- Table top to test assumptions

Applied Learning

Next Week

- As a security team start to meet, plan, code, and learn in the open within your company
- Start meeting with the engineering leads to lay groundwork for collaborative efforts on longer term initiatives
- Write down the 3-5 things that scare you most about working in security at your company
    - This is good doomsday scenario material for later

3 Months Out

- Set 3-5 OKRs for your security program, shoot for quarterly or halves
- Document ***and circulate*** a handful of secure requirements for products and the business as a whole
- Collectively identify and implement several starter metrics to track and display (security + engineering) such as:
  - Your company top 10 vulnerabilities
  - Time to remediate vulnerabilities (by risk)
  - Critical vulnerabilities known per 10kloc
- Implement automated security tests/checks in small pieces at a time

6 Months Out

- Run a table top exercise to stress test the progress made thus far
  - Document the results and the necessary TODOs that come out of the activity
- From your company top 10, select a class of vulnerability to attempt to eradicate through library/framework/service engineering
- Measure progress on OKRs

Questions?

Robert Wood

bob@nuna.com

@robertwood50

Some Reference Materials

- http://www.slideshare.net/zanelackey/building-a-modern-security-engineering-organization
- http://www.bishopfox.com/blog/2015/03/beyond-security-requirements-secure-requirements/
- http://www.slideshare.net/michael_coates/shape-developer-firstsecurity
- http://www.slideshare.net/StephendeVries2/continuous-security-testing-with-devops
- https://www.owasp.org/images/e/e3/DefenderEconomics.pdf