Strategy Development with Wardley Mapping - Applying Concepts

https://open-security-summit.org/

Agenda

- The 4 Problems
  - The Communication problem
  - The Engineering problem
  - The Management problem
  - The Skills and Structure problem
- Closing thoughts

Not a single map in this talk

https://i.ytimg.com/vi/ZlEcKCm3gbE/maxresdefault.jpg

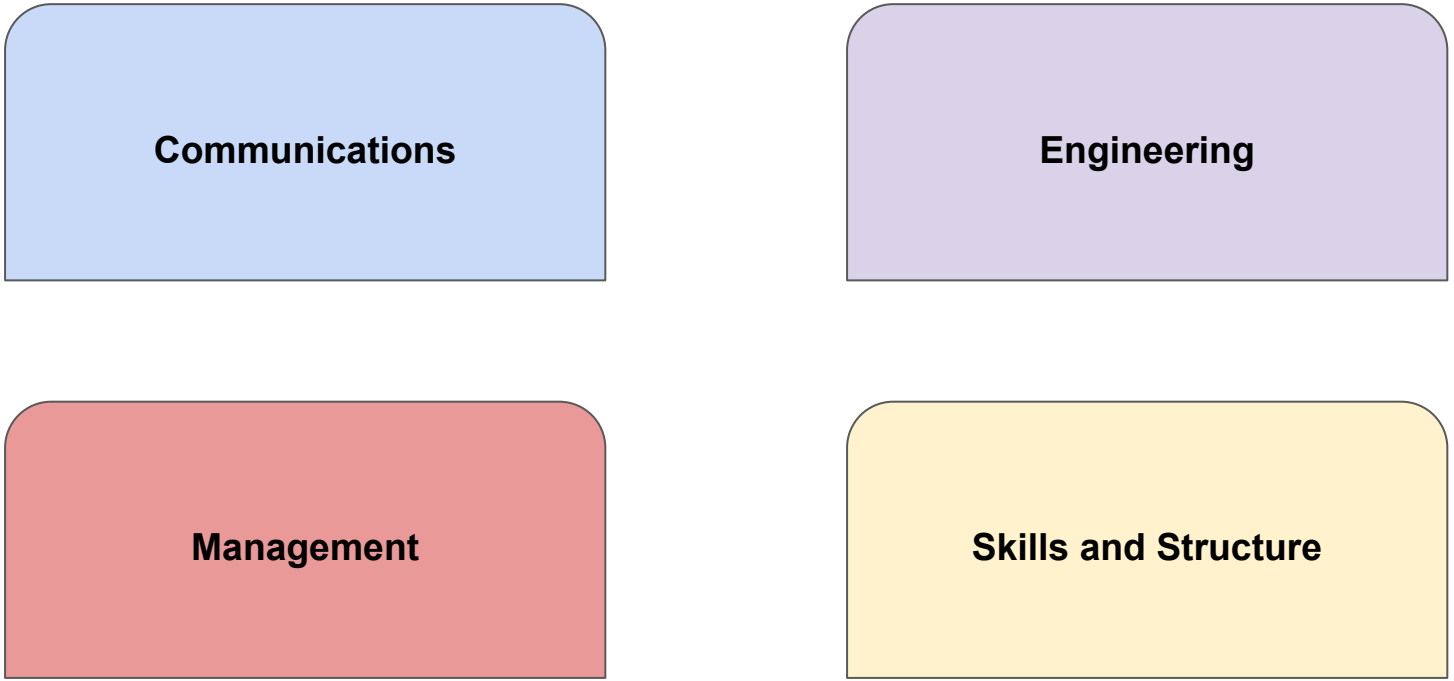
Wardley Map

Wardley Mapping

Applying Climatic Patterns and Doctrine as a way of assessing our landscape and the forces acting on it in order to help manage and prepare for change and set strategy

“Affordance is what the environment offers the individual” James J. Gibson

The 4 Cyber Security Problems for the next decade



The Communications Problem

- Business struggles to understand security teams or “why they should care” (different views of value consumed and value created)
- Security teams struggle to understand each other

Using Doctrine on the Communications problem

OPEN SECURITY SUMMIT

Adopt common language

- **Use architectural methods which help connect concerns at different levels (SABSA)**
- **Adopt Threat Taxonomy as Categorisation**
- **Adopt Cyber Security Style guide to normalise security language across the business**

Focus on high situational awareness

- Correlation between awareness and performance
- No presence or feedback loops from Development ceremonies
- **Do the work to connect multiple timespans of discretion**
- **Adopt Cyber Defence Matrix**

Challenge assumptions

- Have an open door for Engineers/Devs and Security staff to challenge ways of doing things and complying
- Welcome and embrace challenge
- Ask more, dictate less
- Give others agency

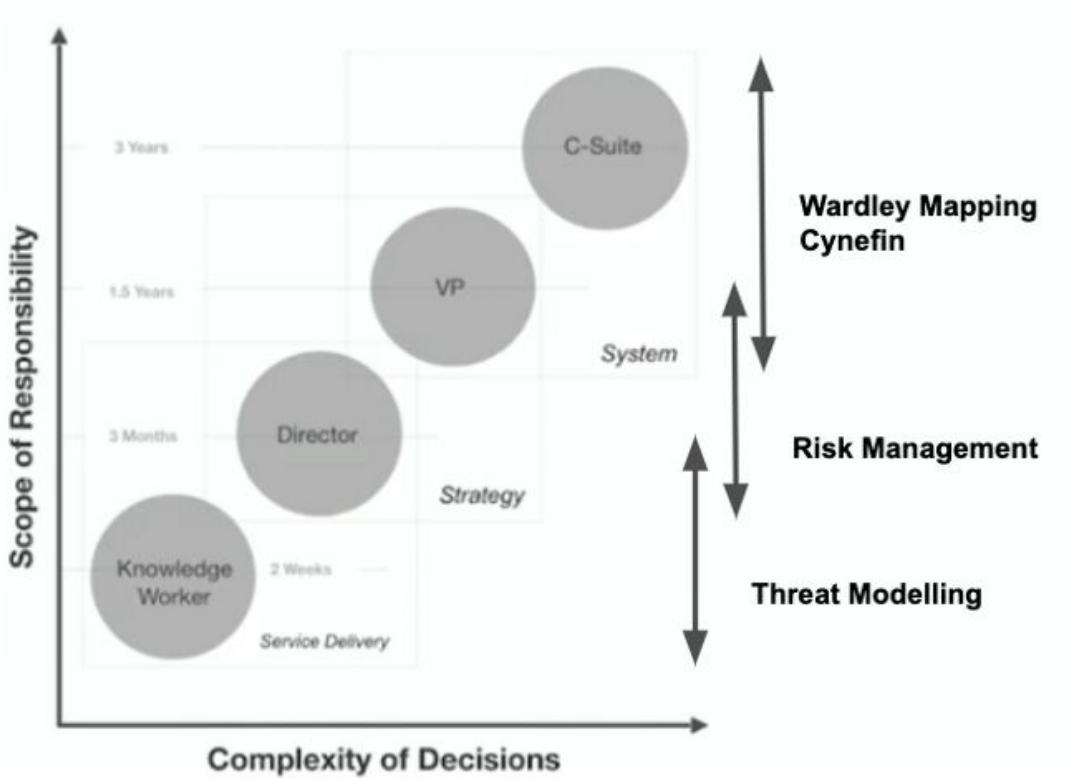
Using Climate on the Communications problem

Evolution of communication can increase the speed of evolution overall

- Improving our communication structures and its effectiveness will allow us to evolve quicker, and the quicker we can evolve our security approaches to better serve our organisations

Communications problem - Focus on high situational awareness

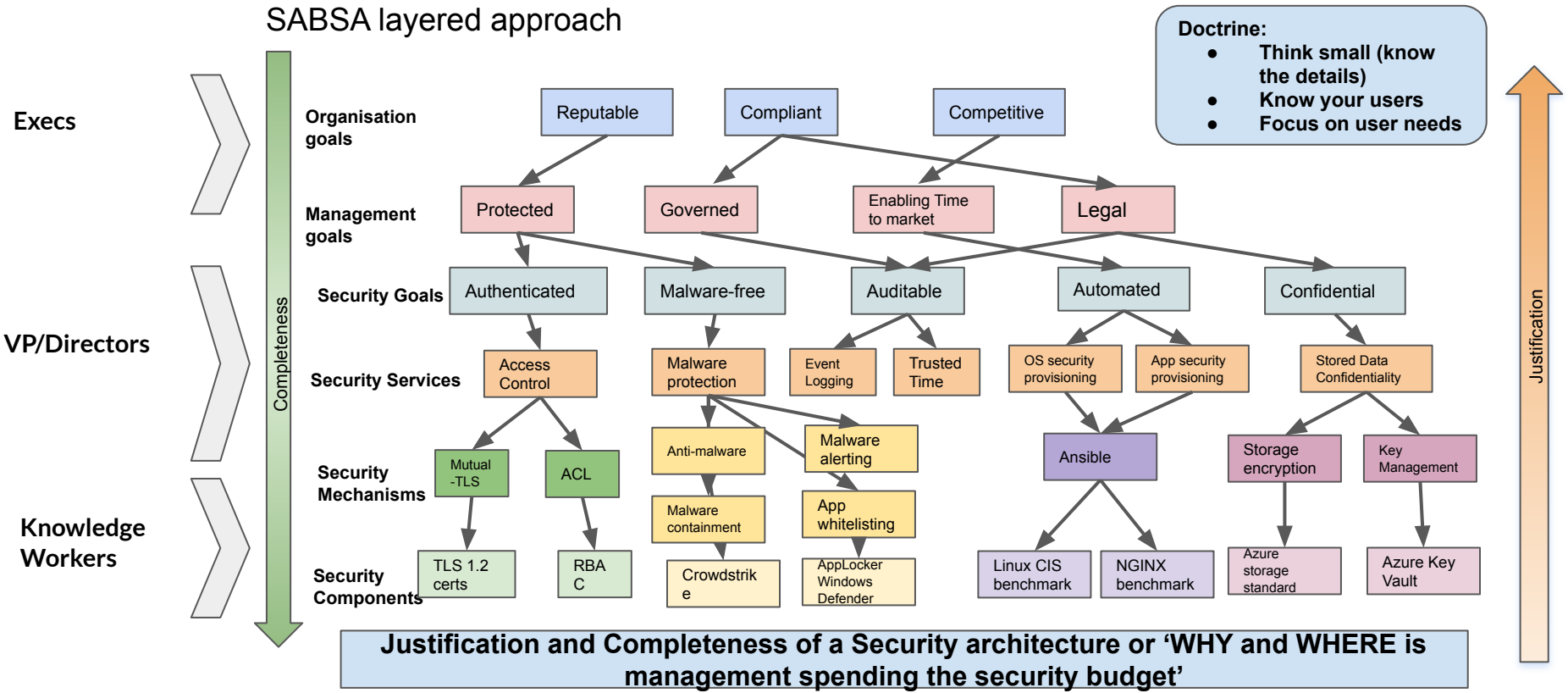
https://medium.com/@marioplatt/social-practices-and-timespan-of-discretion-in-cyber-security-cef4fdc16663



Based on @cyetain Jabe Bloom's "Whole work, Sociotechnicity of DevOps
https://www.youtube.com/watch?v=WtfncGAeXWU

Communications problem - Focus on high situational awareness

Adopt common language - Architectural Traceability



Adopt common language

Engineering
Tools, trade-offs, constraints
Tech Jargon

Anna, the Tech Gal

Business value of assets
Business value chains
Market forces
Shareholder pressures
Business attributes

Carl, the Security Manager

Sales-speak
Gartner-infused buzzwords
FUD
Specialist Jargon

Peter, the Security Product Specialist

Recovery procedures
SLAs
Incident Indicators of Compromise

Sushma, the Incident Responder

Control testing
Evidence review
Testing procedures
Checklists and spreadsheets

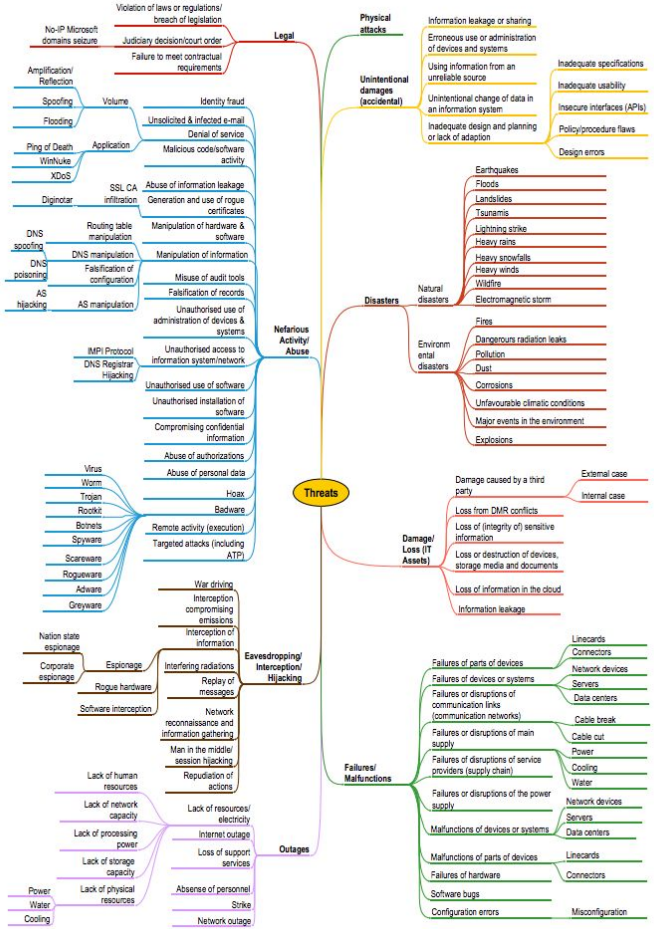
Felipe, the Compliance dude

Icons from https://commons.wikimedia.org/

Adopt common language - Threat Taxonomy

Collecting

Sorting
Consolidating



Asset exposure

Vulnerability exploitation

Adopt common language - Threat Taxonomy

Handling a phishing incident (ENISA Threat Taxonomy)

Icons from https://commons.wikimedia.org/

Anna, the Tech Gal

Carl, the Security Manager

Carl, the Security Product Specialist

Sushma, the Incident Responder

Felipe, the Compliance dude

Adopt common language - Threat Taxonomy

Handling a phishing incident (ENISA Threat Taxonomy)

Icons from https://commons.wikimedia.org/

Anna, the Tech Gal

Carl, the Security Manager

Peter, the Security Product Specialist

Sushma, the Incident Responder

Felipe, the Compliance dude

Incident
Contact Centre user clicked on malicious link and host got malware infected
Threat: Malicious Code *Threat: Receive unsolicited email*

Vulnerability
Mail filter didn't detect malware because it was outdated
Threat: Erroneous administration of devices

Corrective Control
Update filter with latest signatures
Threat: Erroneous administration of devices

Adopt common language - Threat Taxonomy

Handling a phishing incident (ENISA Threat Taxonomy)

Icons from https://commons.wikimedia.org/

Anna, the Tech Architect Gal

Carl, the Security Manager

Peter, the Security Product Specialist

Sushma, the Incident Responder

Felipe, the Risk&Compliance dude

Vulnerability
Host AV didn't detect or contain malware
Threat: Inadequate adaptation

Vulnerability
Mail filter didn't detect malware because it was outdated
Threat: Erroneous administration of devices

Corrective Control
Improve Operational process on mail filter management
Threat: Inadequate design and planning

Adopt common language - Threat Taxonomy

Handling a phishing incident (ENISA Threat Taxonomy)

Icons from https://commons.wikimedia.org/

Anna, the Tech Architect Gal

Carl, the Security Manager

Peter, the Security Product Specialist

Sushma, the Incident Responder

Felipe, the Risk&Compliance dude

Risk
Operators can't perform Customer Service function when systems are being recovered
Threat: Loss of support services

Risk
Failure to meet clause 12.4 of Standard contractual clauses, affecting 120 contracts
Threat: Failure to meet contractual requirements

Risk
Customer service function loss leads to breach of contractual SLA with customers
Threat: Failure to meet contractual requirements

Adopt common language - Threat Taxonomy

Handling a phishing incident (ENISA Threat Taxonomy)

Icons from https://commons.wikimedia.org/

Anna, the Tech Architect Gal

Carl, the Security Manager

Peter, the Security Product Specialist

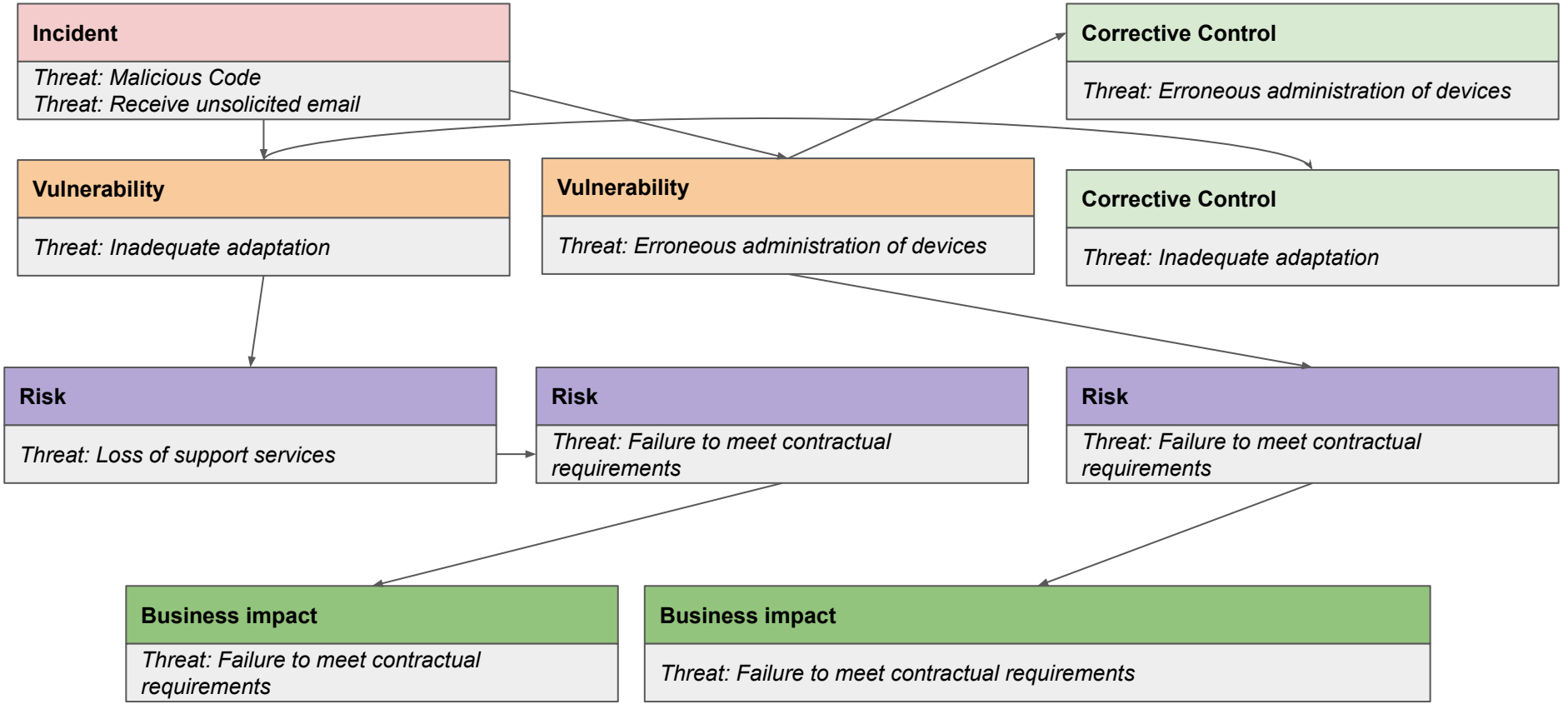
Sushma, the Incident Responder

Felipe, the Risk&Compliance dude

Business impact
6h outage will result in £60.000 in losses due to service credits compensation to customers
Threat: Failure to meet contractual requirements

Business impact
Failure to keep appropriate security measures may result in cancellation of up to 7 contracts, resulting in net loss of £600M over 3 years
Threat: Failure to meet contractual requirements

Threat Taxonomy for the Phishing incident



Adopting Cyber Security Style Guide

CYBERSECURITY STYLE GUIDE

Over the years, we've compiled a style guide to keep ourselves technically accurate and up to date in our reports, presentations, and social media interactions. Now we want to share our current standards with you.

Each term earned its place by being unintuitive in some way:

- It may look like a non-technical word (execute, pickling, shell)
- It may be uniquely written (BeEF, LaTeX, RESTful)
- It may not follow a clear pattern (web page vs. website)
- It may have a very specific technical distinction (invalidated vs. unvalidated)
- Or its meaning may change depending on the context (crypto, PoC, red teaming)

https://www.bishopfox.com/cybersecurity-style-guide/

Adopting Cyber Security Style Guide

OPEN SECURITY SUMMIT

malicious actor (n.)

This represents a wide range of potential attackers from individuals to nation-states. When writing about individual threats, use attacker or malicious user.

incident response (IR) plan (n.)

Use lowercase when writing about the concept of the plan. Capitalize it if referring to the name of a specific document.

Related: **IR plan**, **SIR plan**

0-day (n. or adj.)

A "zero-day" or "oh-day" finding. In formal writing, it's better to use zero-day finding, previously undisclosed vulnerability, or publicly undisclosed vulnerability.

abuse (v.)

This verb is OK in set phrases but do not use it on its own. Try alter, automate, compromise, deface, exhaust, exploit, force, impersonate, intentionally misuse, manipulate, reuse indefinitely, take advantage of, or a context-specific verb.

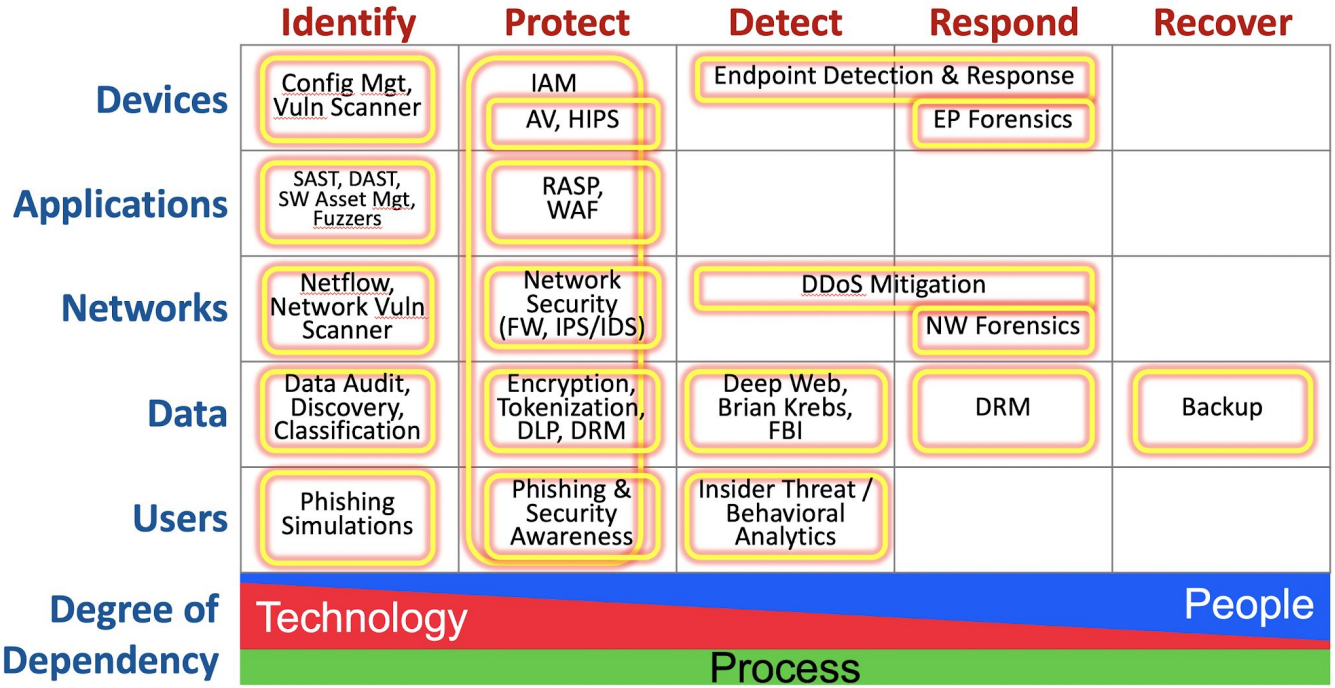
https://www.bishopfox.com/cybersecurity-style-guide/

The Cyber Defense Matrix

	Identify	Protect	Detect	Respond	Recover
Devices					
Applications					
Networks					
Data					
Users					
Degree of Dependency	Technology / People / Process				

https://www.rsaconference.com/industry-topics/presentation/cyber-defense-matrix-reloaded

Enterprise Security Market Segments

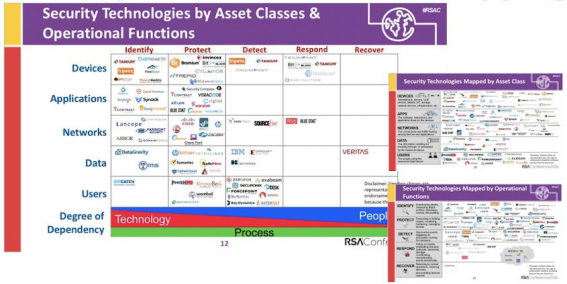


https://www.rsaconference.com/industry-topics/presentation/cyber-defense-matrix-reloaded

Recap from 2016 Briefing

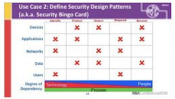
https://bit.ly/cdm-rsa2016

Primary Use Case: Vendor Mapping



Other Use Cases

Differentiating Primary & Supporting Capabilities



Defining Security Design Patterns

Maximizing Deployment Footprint

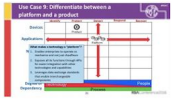
Understanding the New Perimeter

Calculating Defense-in-Depth

Balancing Your Portfolio Budget

Planning for Obsolescence

Disintermediating Security Components



Comparing Point Products vs Platforms

Finding Opportunities for Automation

Identifying Gaps in People, Process, Tech

https://www.rsaconference.com/industry-topics/presentation/cyber-defense-matrix-reloaded

The Engineering Problem

- Security standards turned into fads, when we should be focusing on Engineer/Developer Experience
- Bolt-on security vendor driven security adds cost, complexity and promotes silo-ed thinking and prevents agency from technical teams. This decreases resilience and introduces business risk
- Most still trying to secure microservices and orchestrated systems like they're on-prem n+1 architectures

Understanding Engineering problem through Climate

OPEN SECURITY SUMMIT

Components can co-evolve (practice with activity)

- Adoption of loose-coupled microservices reduces the scope of what teams are securing and their delivery methods
- What would need to happen for you to trust their outputs ?
- The world is now declarative. Do your security processes reflect that ?

No choice over evolution

- We risk becoming the bottlenecks (alongside the other gatekeepers)
- Our competitors are likely adapting. Move early for strategic advantage, or move late to catch up. You choose
- We'll struggle to retain talent (both Security and Engineering) if we don't evolve

Competitors actions will change the game

- Don't just think about your immediate comfort. Think 5 years down the line
- **Think about the cost of security and your business's bottom line**
- Business enablement > Ticking all the Gartner boxes

Using Doctrine on the Engineering problem

OPEN SECURITY SUMMIT

Use a systematic mechanism of learning

- RULE #1 - If you have an old-school team, this is not the time to SPEAK, it's time to LISTEN
- Do you even Docker bro ? Identify your team's knowledge gaps
- Cross-train within the team. Get people to teach each other
- Ask for permission to see a day in the life of Engineer, and do it regularly

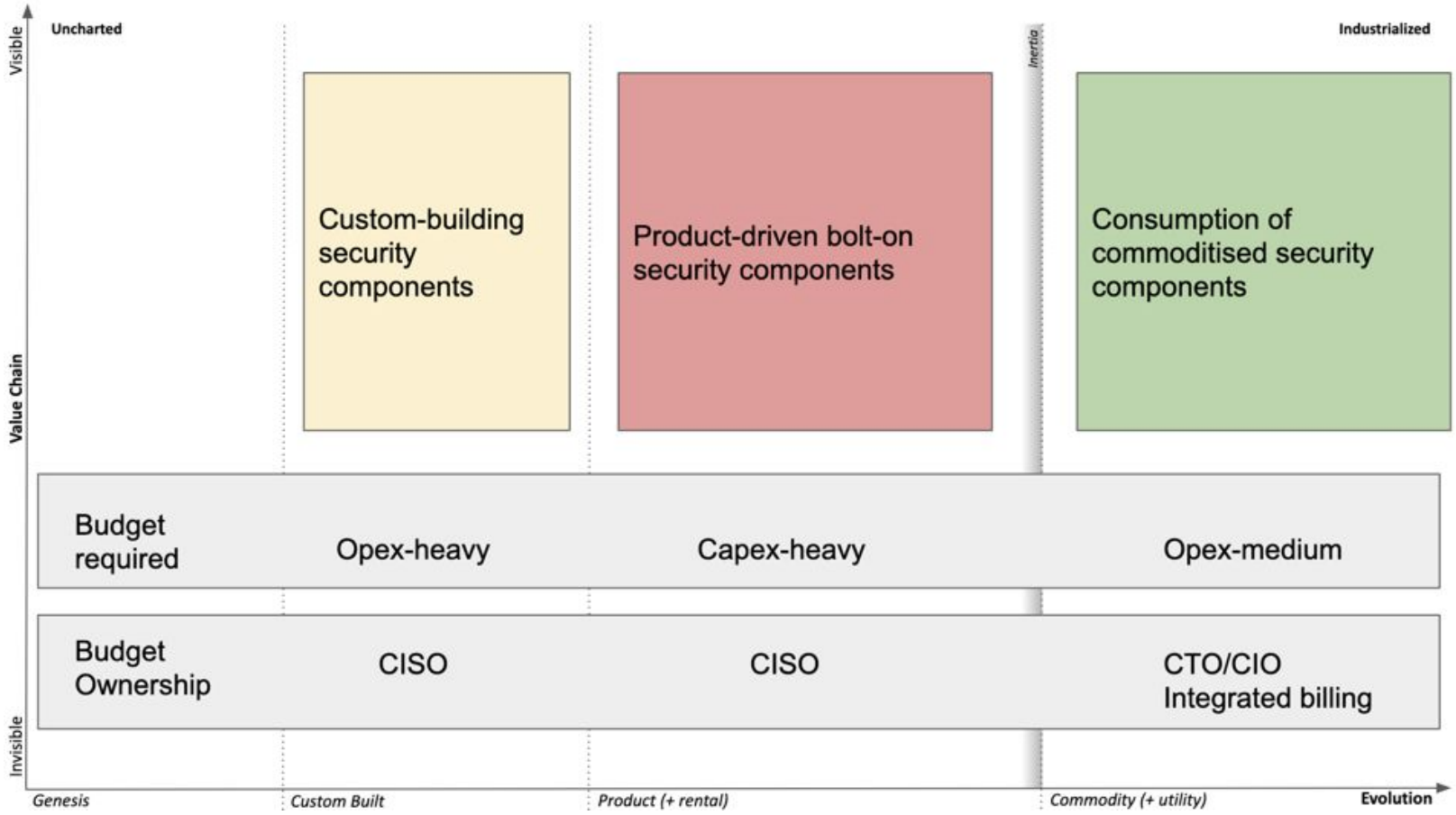
Use appropriate methods

- Remember your associated practices will be in Genesis (independent of evolution of component)
- No gatekeeping, help Guardrail
- Express what would make you trust more
- **Avoid anti-patterns**
- **Aim for Resilience**
- **Learn from Safety Engineering**

Use appropriate tools

- The tools need to be in the hands of Engineers and Devs
- Should you even be choosing tools ?
- Are the tools ready for CI/CD and Engineer agency ?
- Use commodities & native where possible

No choice over evolution

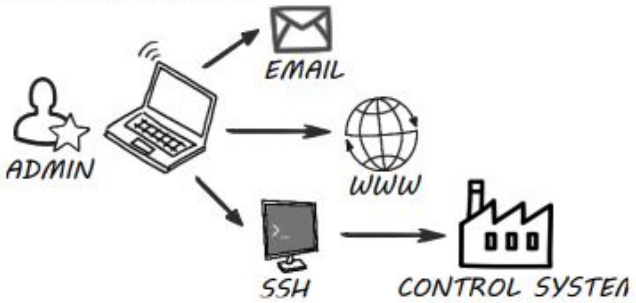


Avoid Security anti-patterns

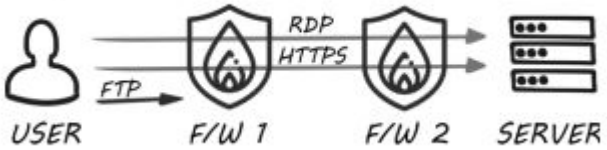
https://www.ncsc.gov.uk/whitepaper/security-architecture-anti-patterns

OPEN SECURITY SUMMIT

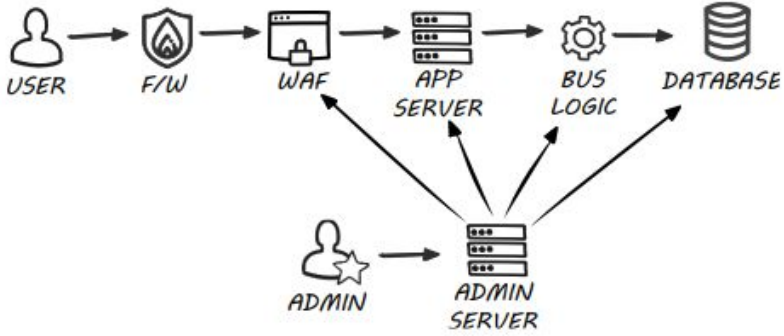
Anti-pattern 1: 'Browse-up'



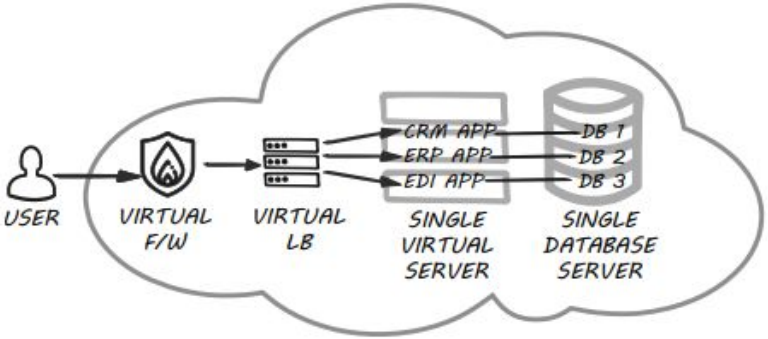
Anti-pattern 3: Back-to-back firewalls



Anti-pattern 2: Management bypass



Anti-pattern 4: 'On-prem' solution



Current Security Analysis

"When you ask an engineer to make your boat go faster, you get the trade-space. You can get a bigger engine but give up some space in the bunk next to the engine room. You can change the hull shape, but that will affect your draw. You can give up some weight, but that will affect your stability. When you ask an engineer to make your system more secure, they pull out a pad and pencil and start making lists of bolt-on technology, then they tell you how much it is going to cost."

- Prof Barry Horowitz, UVA

Aim for Resilience

"It's something a system (your organisation, not your software) ***does****, not what it* ***has.***

Resilience is sustained adaptive capacity, or ***continuous adaptability to unforeseen situations****" John Allspaw*

https://devopsdays.org/events/2019-washington-dc/program/john-allspaw/

https://github.com/lorin/resilience-engineering

Resilience in Security 101 (Kelly Shortridge)

OPEN SECURITY SUMMIT

Robustness = withstanding and resisting a negative event. Think of this as your systems' ability to prevent or block pwnage.

Adaptability = managing your current systems' response to negative events. Think of this as minimizing event impact in your systems, optimizing for system recovery, and minimizing friction involved in changing your systems.

Transformability = reorganizing your system in response to updated assumptions. Think of this as moving to a new system configuration or trajectory when existing conditions are indefensible.

@swagitda_ https://swagitda.com/blog/posts/resilience-in-security-101/

Resilience in Security 101 (Kelly Shortridge)

OPEN SECURITY SUMMIT

Robustness = withstanding and resisting a negative event. Think of this as your systems' ability to prevent or block pwnage.

Adaptability = managing your current systems' response to negative events. Think of this as minimizing event impact in your systems, optimizing for system recovery, and minimizing friction involved in changing your systems.

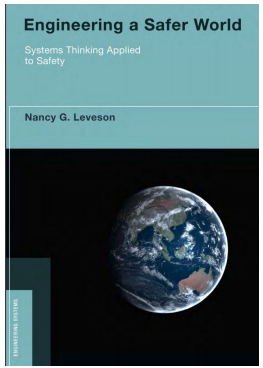
Transformability = reorganizing your system in response to updated assumptions. Think of this as moving to a new system configuration or trajectory when existing conditions are indefensible.

@swagitda_ https://swagitda.com/blog/posts/resilience-in-security-101/

Learning Safety

Human factors concentrates on the "screen out"

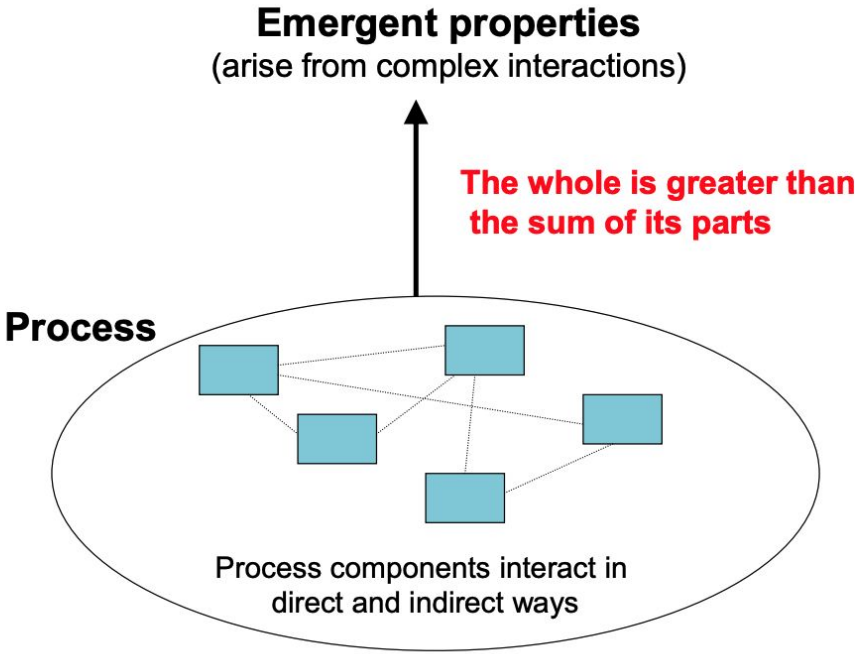
Engineering concentrates on the "screen in"



Not enough attention on integrated system as a whole

Nancy Leveson, http://sunnyday.mit.edu/workshop2019/STAMP-Intro2019.pdf

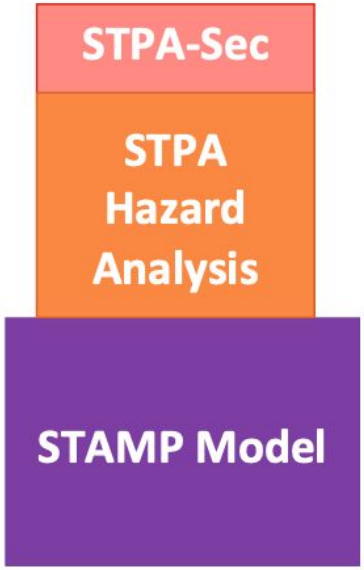
Both Safety and Security are Emergent properties



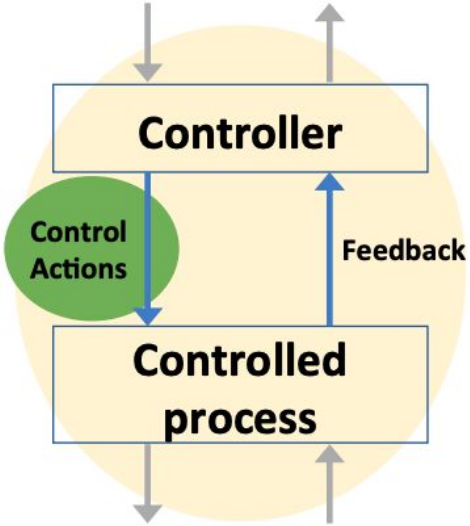
Safety and security are emergent properties

Nancy Leveson, http://sunnyday.mit.edu/workshop2019/STAMP-Intro2019.pdf

STPA-Sec Extends STPA



- **Define system purpose and goal**
- **Identify accidents and hazards**
- **Draw the control structure**
- **Step 1: Identify unsafe/unsecure control actions**
- **Step 2: Identify causal scenarios**
- **Wargame**



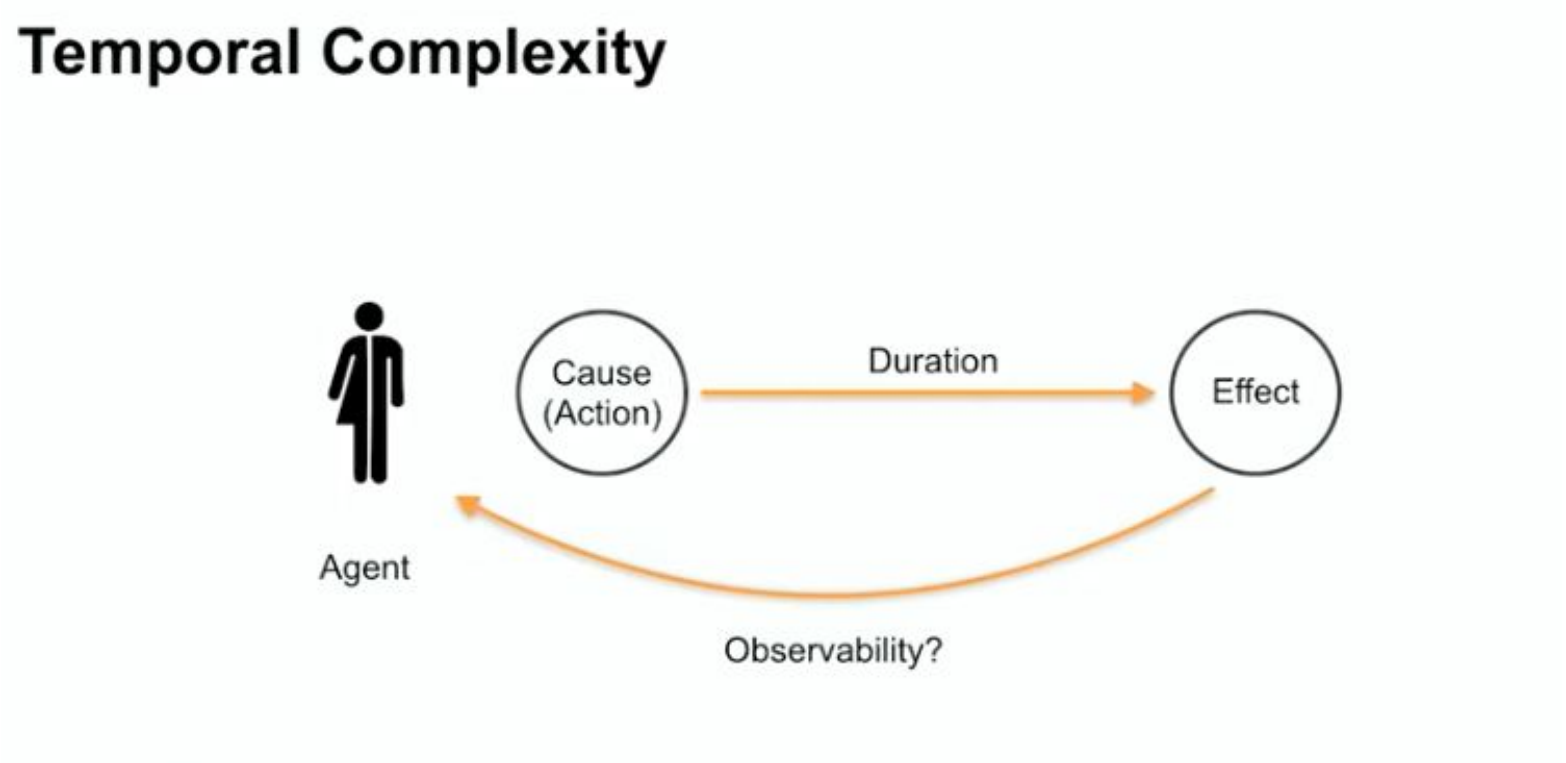
WYOUNG@MIT.EDU

William Young Jr ,
https://psas.scripts.mit.edu/home/wp-content/uploads/2017/04/STAMP_2017_STPA_SEC_TUTORIAL_as-presented.pdf

The Management Problem

- Millionaire CISO budgets has made it commonplace and acceptable to overspend on security
- Short tenures from CISOs implies lack of skin-in-the-game and preference for short term minded decision making
- Modern development practices require strategic focus on providing agency to others, as opposed to the empire building of the last decade
- Security programmes over rely on self-assessment of needs, and not on enough on taking advantage of business affordances
- Qualitative-only approaches to managing risk is bias-on-a-matrix

Most CISOs don’t get to see effects of their decisions



Jabe Bloom @cyetain
https://www.youtube.com/watch?v=WtfncGAeXWU

https://medium.com/@marioplatt/social-practices-and-timespan-of-discretion-in-cyber-security-cef4fdc16663

Understanding Management problem through Climate

Success breeds inertia

- Relationships with vendors and integrators will hold you back
- Familiarity with how to engineer and support models, makes acquiring Products a comfortable position
- LEARN.TO.LET.GO

Inertia increases the more successful the past model is

- The Command and Control, Gatekeeper approach is no longer what your organisation needs
- Stop measuring yourself (and your peers) on how big your capital budget or your team size is
- It's not just about the role you do now. Does your next employer require the skills you have or do they different skills you're not yet qualified to perform ?

Using Doctrine on the Management problem

OPEN SECURITY SUMMIT

Be pragmatic

- Ensure that the job is being done, not who is doing it
- Knowledge may be inadequate. Train your teams

Think Fast, Inexpensive, Restrained and Elegant

- Don't commit millionaire budgets up-front
- Start with Open Source. You may not need more
- In CI/CD, use Docker for security testing

Effectiveness over efficiency

- Your processes likely create tension. Don't make the ineffective more efficient
- Delegate. Find secure ways to stop being in the way. Spot-audit instead
- Collaborate - Facilitate - Provide silent service

Manage inertia

- **Understand root cause of inertia to change (existing practice, power structures, previous investment, political capital, training) and work within affordances**

Bias towards the new (be curious, take risks)

- **Experiment, and take the opportunity to collaborate in parallel with multiple teams**
- Choose a few projects to change your products for a commodity

Inertia due to success of past model

But wait! How are these "security" solutions?

BLOCKCHAIN

docker

Distributed

DDoS Resistant

The best solution against a distributed attack is a distributed service

24 7

Availability

Immutable

Changes Easier to Detect and Reverse

Unauthorized changes stand out and can be reverted to known good

Integrity

Ephemeral

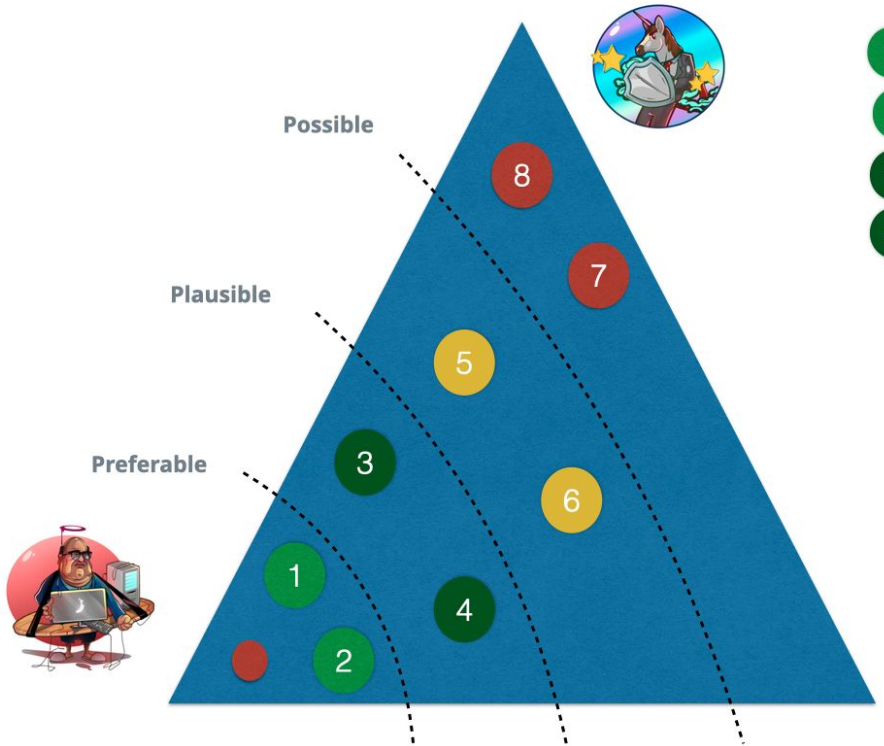
Drives Value of Assets Closer to Zero

Makes attacker persistence hard and reduces concern for assets at risk

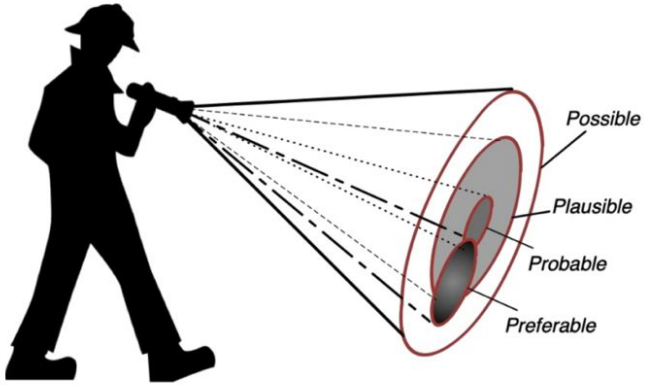
Confidentiality

https://www.slideshare.net/sounilyu/distributed-immutable-ephemeral-new-paradigms-for-the-next-era-of-security

Meet them where they are



1. Pentest
2. Code review
3. Threat modelling
4. Policy-as-Code
5. Integrated DAST/SAST
6. GRC integration
7. Blameless post-mortems
8. Chaos Engineering



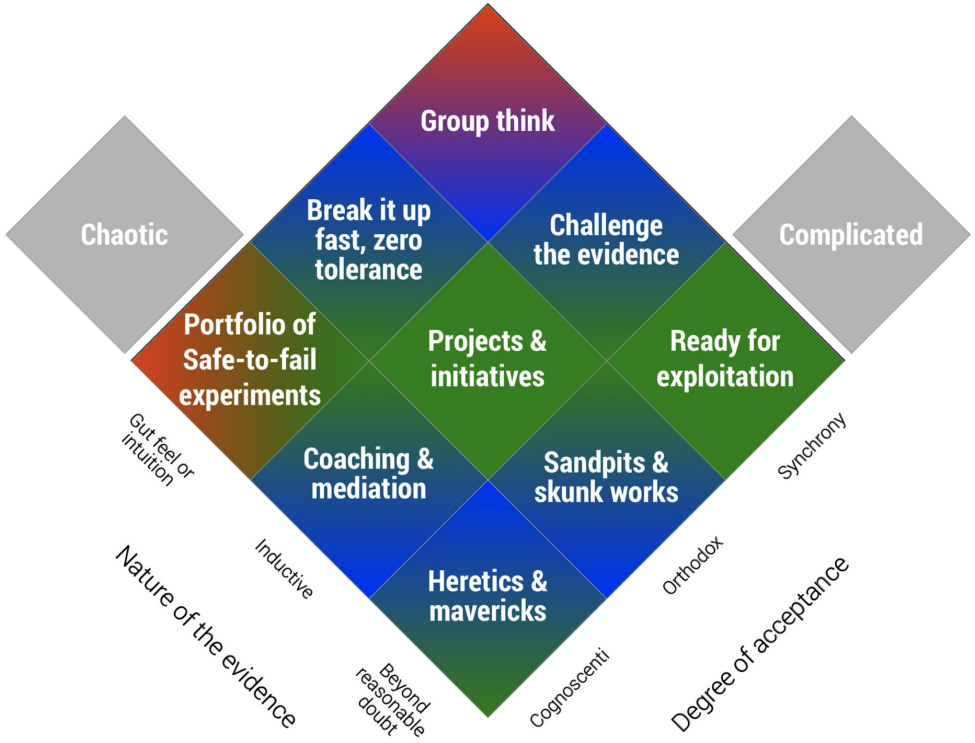
Constraint is TRUST in both teams and process

Jabe Bloom @cyetain https://www.slideshare.net/cyetain/three-frames-devopsdays-atl

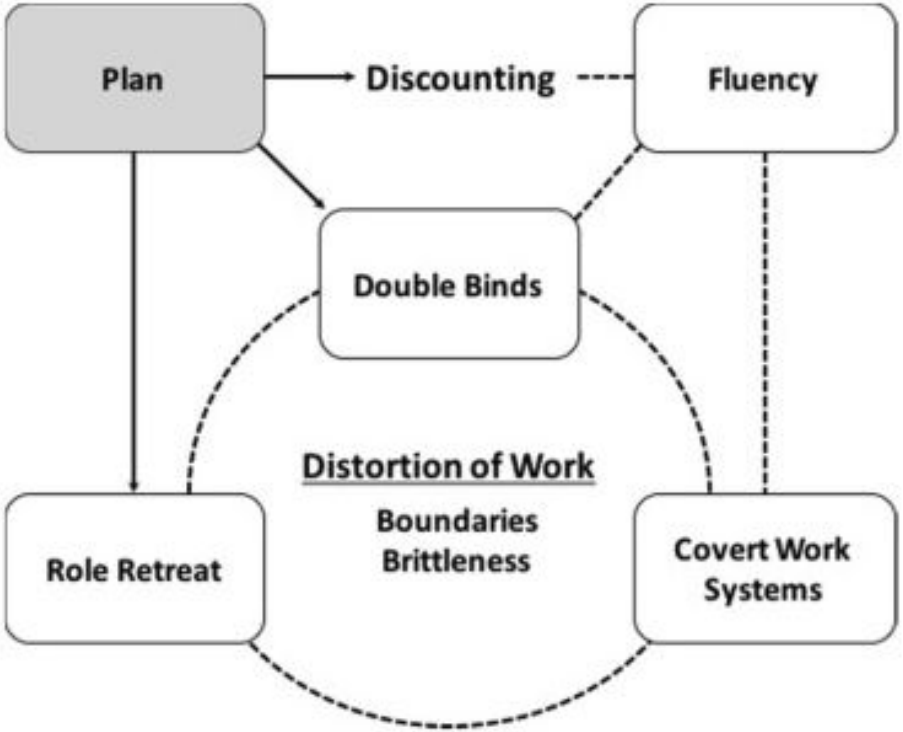
Managing Complex problems

OPEN SECURITY SUMMIT

Complex domain model

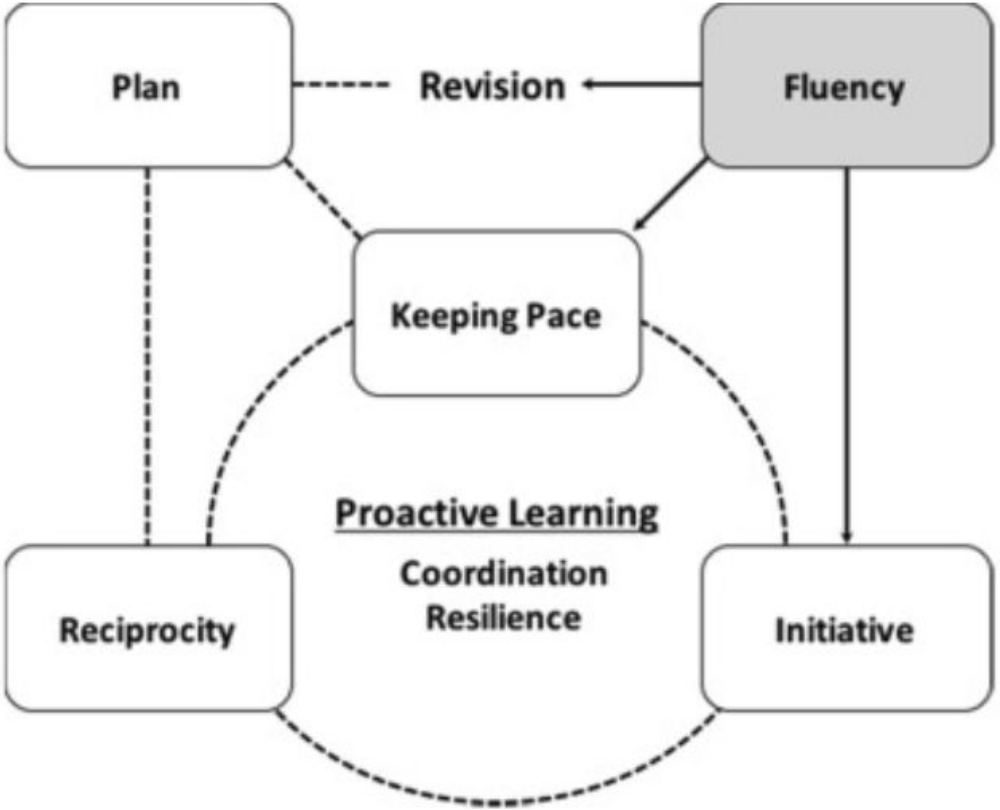


What isn't working - Learn from Safety II - Centralized control



David Provan et al https://www.sciencedirect.com/science/article/pii/S0951832018309864

What is working - Learn from Safety II - Guided adaptability



David Provan et al https://www.sciencedirect.com/science/article/pii/S0951832018309864

The Skills and Structure problem

- Static Org-chart thinking enacts significant barriers to communication with Product and Platform teams
- We cannot pragmatically expect all Engineers to become security experts, and we need to design for cognitive load
- Our Security teams may not have the skills we need to be successful over the next decade
- Stop trying to find unicorns. We need to be able to train for the skills we require to be successful in our context
- And also to leverage the current skills we have and work within those constraints with adjacent possibles

Using Climate on the Skills & Structure problem

OPEN SECURITY SUMMIT

Components can co-evolve (bias towards data)

- It wasn't just the tech which changed. The Social practices and governance patterns did too
- **Your org chart is killing you and will keep you down permanently**

No choice over evolution (Red Queen)

- The Command and Control, Gatekeeper approach is no longer what your organisation needs from us
- We design for agility and agency. Our businesses can no longer afford to choose between quality and speed
- **Be a learning organisation, or your competition will**

Creative Destruction

- **2 choices: Adapt and learn for yourself, or wait for the business to bring someone else to do it for you. Then you'll have 2 problems**
- **The good news, is that old-school Infosec teams know is still valid, so it's about leveraging and not putting in the bin**
- The market is slowly eroding the need for "legacy security skills"

Using Climate on the Skills & Structure problem

OPEN SECURITY SUMMIT

Focus on User needs	Distribute power and decision making	Think aptitude and attitude
• We provide a service to the Exec teams and to our front lines (digital or not). Let's understand what THEY need from us • Their preferences and requirements, should be our constraints (not the reverse)	• Less checklists and processes, more heuristics and rituals • Knowledge is power, educate but don't overwhelm • Be more deliberate in Product level support, let's talk less about Organisational perspectives • Product owners need to be allies. Foster those relationships, aim for level playing field	• Not Pioneers-Settlers-Town Planners. **You're not ready for that** • Focus on acquiring the skills you require to be successful • If hiring new, remember it's easier to teach a Dev/QA engineer about security than to change an old-world Infosec person's view of the world • More diversity, less "inbreeding" in Infosec, please

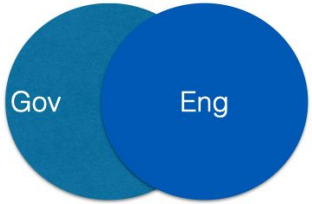
Conway's Law

"Organisations which design systems are constrained to produce designs which are copies of the communication structures of these organisations" Conway

"If the architecture of the system and the architecture of the organisation are at odds, the architecture of the organisation wins" Ruth Malan

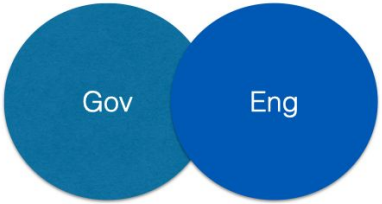
Your org chart is holding you back

‘Team Topologies’ applied to Cyber Security



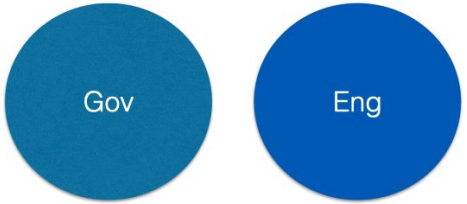
Collaboration

- Governance mapping (metadata management)
- Joint Threat modelling and Risk assessments
- Visibility and Reporting
- Management system updates



Facilitation

- Focus on cross-team training and contextual guidance
- Separate Threat modelling and Risk assessment sessions
- Boundary spanning more active between domains



X-as-a-Service

- Fully defined interfaces (Team or actual APIs)
- Informed/Supportive roadmaps
- Exception Management

Team Evolutionary Path

Be a learning organisation

OPEN SECURITY SUMMIT

“Learning organisations become graduate studies in the skills they require to be successful “
Andrew Clay Shaffer @littleidea

Creative Destruction - with a pinch of leverage

Traditional SDLC → Agile / Scrum → DevOps → DevSecOps

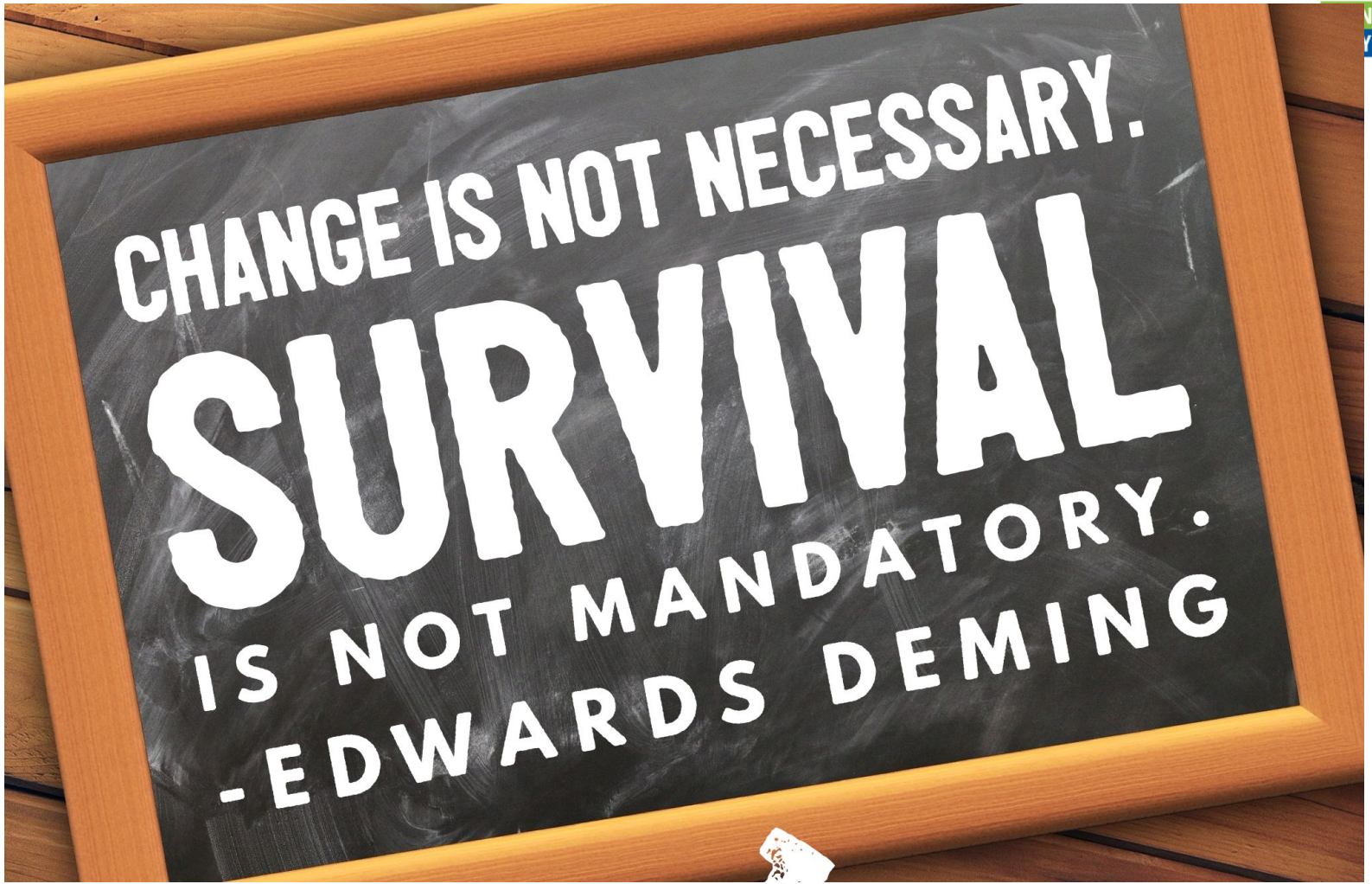
Requirements-as-s preadsheets → User Stories → Scenario Testing → Compliance -as-Code

Qualitative Risk Management → Qualitative Risk Management with Quantitative analysis → Quantitative Risk Management (FAIR)

Risk Analysis → Threat modelling → Threat modelling as code

Security Products → Cloud-capabilities / CNCF

Robust Engineering → Resilience Engineering


CHANGE IS NOT NECESSARY.
SURVIVAL
IS NOT MANDATORY.
-EDWARDS DEMING

Solving Cyber Security in your organisation in the next 5 years (extending on Sounil Yu's hypothesis)

OPEN SECURITY SUMMIT

Communications Problem

- Threat Taxonomy as Categorisation
- Style Guides
- SABSA Security Architecture
- Cyber Defence Matrix

Engineering Problem:

- Avoid anti-patterns
- Aim for Resilience
- Learn from Safety Engineering
- Less Gatekeeping, more Guardrailing (with listening)

Management Problem:

- Smarter spend / less suppliers
- Apply Doctrine
- Address sources of inertia
- Take risks and experiment

Skills and Structure Problem:

- Dynamic and fluid team structures (meet needs)
- Be a learning organisation
- Hire for aptitude and attitude
- Leverage existing skills but deliberately evolve them

Wardley mapping as shared language for meaningful conversations

Q&A

Mario Platt

mario@practical-devsecops.com

Twitter: @madplatt
LinkedIn: marioplatt
Medium: @marioplatt