RSA Conference 2017

Learning Labs

Threat Modeling Demystified

Brook S.E. Schoenfield

Author, Principal Engineer, Lead Product Security Architect, Intel Security

Curious questioner

Table of Contents

Introduction

Many thanks to the people who spent 2 hours with me, Wednesday morning at RSAC 2017, San Francisco. Without your active participation, learning to threat model is a great deal harder.

It turns out, after years of trying to help others climb into the art of seeing systems through the eyes of attackers that groups are one of the best ways to learn. Indeed, threat modeling is best done by the entire development team, anyway. So, by learning in groups, we model the way we are likely to get our best results when analyzing our own systems.

Spending a few hours with people in this activity, or even an entire day, is one of the most rewarding activities that I am privileged to do. Thank you for taking the journey with me.

In this document, I hope to summarize the main points we covered, as well as top provide just a little bit more than the slides for the Lab. I did fill in the notes for most of my slides as well, so they should be a useful reference, too.

Threat Modeling Is Design-Time Analysis

We have a Secure Design Problem

While some security issues (vulnerabilities) are implementation errors, a fair number of security issues are the result of a lack of security thinking while creating the structure of software (architecture) and then expressing the structure and the requirements in a codable/implementable design.

Several obvious design issues were presented, most notably:

- The hacked Jeep Wrangler (failed low level access control)
- A pacemaker that joins open WiFi (open communications that assume all communications are benevolent)
- The Target stores payment terminal hack (failure to isolate sensitive assets and to grant only required access)

Threat Modeling Is Applied to an Architecture

Architecture is the process of ordering the structure of things (e.g., buildings, cities, software systems, an enterprise, software, anything that is sufficiently complex such that the builders wish to structure it). Architecture uses abstraction in order to understand the system and in order to play with particular aspects of it.

Threat modeling is an analysis technique to foster secure design. A threat model is not a "design". Threat models output security requirements which then must be expressed in the design and then implemented and validated.

Start the threat model at the point when the architecture is sufficiently defined such that most if not all of the major functional components are understood.

Revisit the threat model when the architecture materially changes. That is, revisit when components are added or removed, or lines/forms of communication are added or changed.

Threat models are living documents. That is, the analysis may be done as a point in time, but material changes must be reflected in the threat model. This is particularly important for iterative forms of development.

Threat modeling is compatible with iterative development practices. Iterate the threat model appropriately (whenever there are structural changes).

Threat modeling may be thought of as applied security architecture. Security architecture is the practice of building defense-in-depth based upon considering how attackers will try to use a system in order for the attacker(s) to achieve her/his/their goals.

Secure Software Imperatives

We took the time to define the qualities that secure software exhibits. The following design/architecture imperatives describe the behaviors of “secure software”. We noted that only the first imperative directly deals with implementation errors (commonly called “vulnerabilities”). The other 4 imperatives will be fulfilled through secure design practices.

- Be free from errors that can be maliciously manipulated, ergo, vulnerabilities
- Have the security features that customers require for the intended use cases
- Be self-protective; resist the types of attacks that will be promulgated against the software
- In the event of a failure, software must “fail well”, that is fail in such a manner as to minimize consequences of successful attack
- Install with sensible, “closed” defaults

A Process

Knowledge Required Before the Threat Model

The contributing knowledge domains are "The 3 S's: Strategy, Structures, and Specifications." "The 3 S's" encompass: security strategy for the organization and this system, any infrastructure and security services (structures), and things like data classifications, runtime and execution environment, deployment methods, and the like (specifications):

- Strategy
  - Threat landscape
  - The organization's and system's required risk posture
- Structures
  - Existing and implementable security controls
  - Security infrastructure that does not exist (limitations)
- Specification
  - Data sensitivities
  - The runtime and execution environment
  - How do code and systems get deployed?

There is no book of "correct" answers. Each organization is different.

Who Should Participate in The Threat Model?

At the very least, all the domain experts and someone familiar with the threat landscape that is relevant to the system under analysis and the exploits commonly tried against such systems ("security expertise").

Even better, involve everyone who will help to build the software/system. Simply remind everyone who participates that threat models are very sensitive analyses.

Identifying Important Requirements

As was noted earlier, context matters. Without understanding organizational goals and the goals of attackers, all avenues of attack will appear to be equal. In a universe of limited resources, there will be a strong need to focus in on the attacks that pose the most danger to the organization. Importantly, what risks can be tolerated?

Assessing Risk

Set priorities with risk. How much risk is highly context dependent, and is often unique to an organization.

Risk != vulnerability

Threat != vulnerability
Threat != exploit

Participants were introduced to Just Good Enough Risk Rating (JGERR) which is based on Factor Analysis of Information Risk (FAIR), an Open Group standard. JGERR breaks down an attack into 5 terms, 4 of which can be treated as essentially Boolean for the purposes of rating possible attacks when threat modeling. JGERR does not purport to calculate risk, but rather to quickly identify the important attack possibilities, as well as to deprioritize the less important attacks.

"Credible Attack Vector" (CAV) was defined as one method for analyzing attack types.

CAV == Threat & Exploit & Exposure & Vulnerability

Interrupt any CAV term and you interrupt the kill chain, that is the entire CAV. Defenses should negate at least 1 term in the CAV. Sometimes a defense makes it more difficult to achieve one of the terms. We examined how authentication in some cases provides little prevention, and may be used solely to tie an action to a user's ID. (break the kill chain; eliminate the CAV; lower risk exposure)

* Given 0 < CAV < 1, then risk might be expressed as:
* Risk Rating = CAV * Impact

Still, there is no risk if there is no impact. Impact is used since some attacks are stepping stones towards the attacker's goals.

If an attack offers no additional attacker value to achieve ultimate goals, then the attack will not be used in the wild (though security researchers may be interested). This is a type of low impact situation. We examined the situation where a buffer overflow can only be achieved at very high privilege. In such a case, the exploitation does not provide additional attacker value because the attacker already has control of the operating system.

It is important to think not only about harm to a system and organization, but also about whether exploitation provides attacker value.

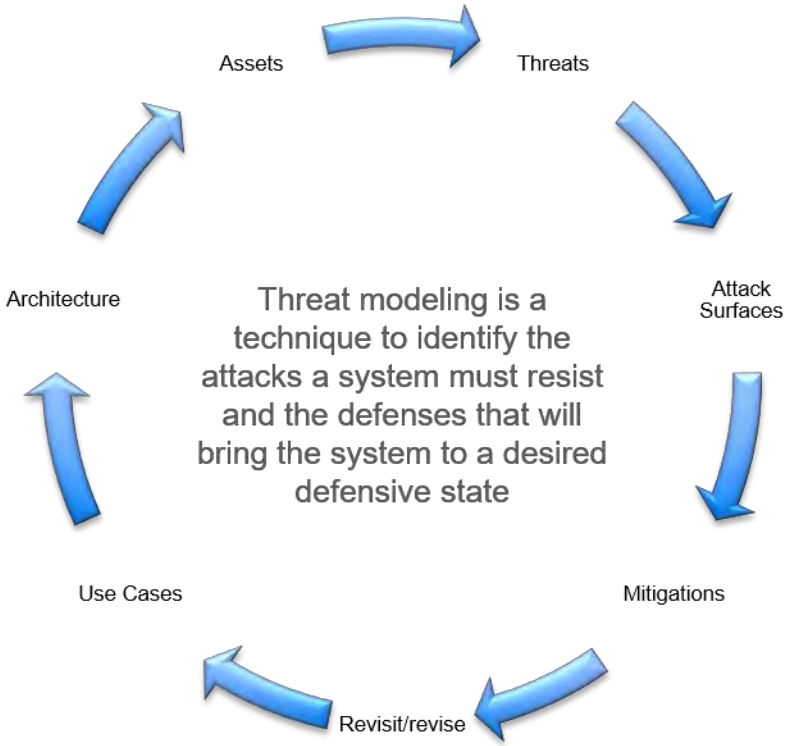
ATASM

I presented threat modeling as a linear progression. But the process is often fractal, as previously uninvestigated components open up further unknown areas where the investigation begins afresh. The analysis may have to recourse back to the start again, complete for that area, then return informed to the area that had been under analysis previously.

Threats ↴
Architecture ↴
Attack Surfaces ↴
Mitigations

Each of these high level activities breaks into sub-steps. It doesn't matter where the analysis starts: assets, use cases, structure, threat landscape, attack surfaces. Each area has to be analyzed for any given system. And each area will have dependencies on the others. How the analysis proceeds seem to be largely a matter of preference and thinking style. Begin where you are comfortable and proceed.

The point is to arrive at a prioritized list of security requirements that will need to be built in order to bring the system to the desired defensive posture.



By Sung Lee; concept by David Wheeler; used with permission

Structure and Flow

There is no perfect architecture view. Component, functional, or logical[1] architecture is often a good place to start. But views can range from the enterprise view ("Conceptual") to detailed physical or technological views. Each view may add important knowledge as different aspects of a complex system are analyzed.

Architecture is at least in part about abstraction to consider an aspect of complexity without getting bogged down in details that aren't relevant for that consideration. Some parts of the structure are highlighted (abstracted from the whole) while details are obscured. Since security

[1] Depending upon the architecture methodology, component, functional, and logical may be distinct views or may be different names for the same view. In some systems, these can also be combined. So I have listed them all in the hopes that one of these names will fit with your local architecture methodology.

operates at all levels and across most if not all domains, for a security analysis, many different views may have to be considered.

A data flow diagram (DFD) can highlight inputs and outputs. Inputs are a great place to identify potential attack points ("attack surfaces"). Still, one must decide at what level of abstraction to draw the system.

Generally, the architecture of the intra-process call tree will be too granular. Still, where untrusted operating system communications or where the operating system itself may be compromised, even intra-process calls may have to be considered. In any event, complex systems generally get considered at their coarsest grain structure, and then sub-views, breaking these large structures into component parts ("factoring" the architecture) may be needed in order to keep views understandable and uncluttered.

Threat Agents

Table 2.1 Summarized Threat Attributes

Threat Agent	Goals	Risk Tolerance	Work Factor	Methods
Cyber criminals	Financial	Low	Low to medium	Known proven
Industrial spies	Information and disruption	Low	High to extreme	Sophisticated and unique
Hacktivists	Information, disruption, and media attention	Medium to high	Low to medium	System administration errors and social engineering

We played with categorizing the goals and capabilities of varying threat agents (human actors who attack computer systems). Each work group created their own list from which to work.

A Threat Agent Matrix

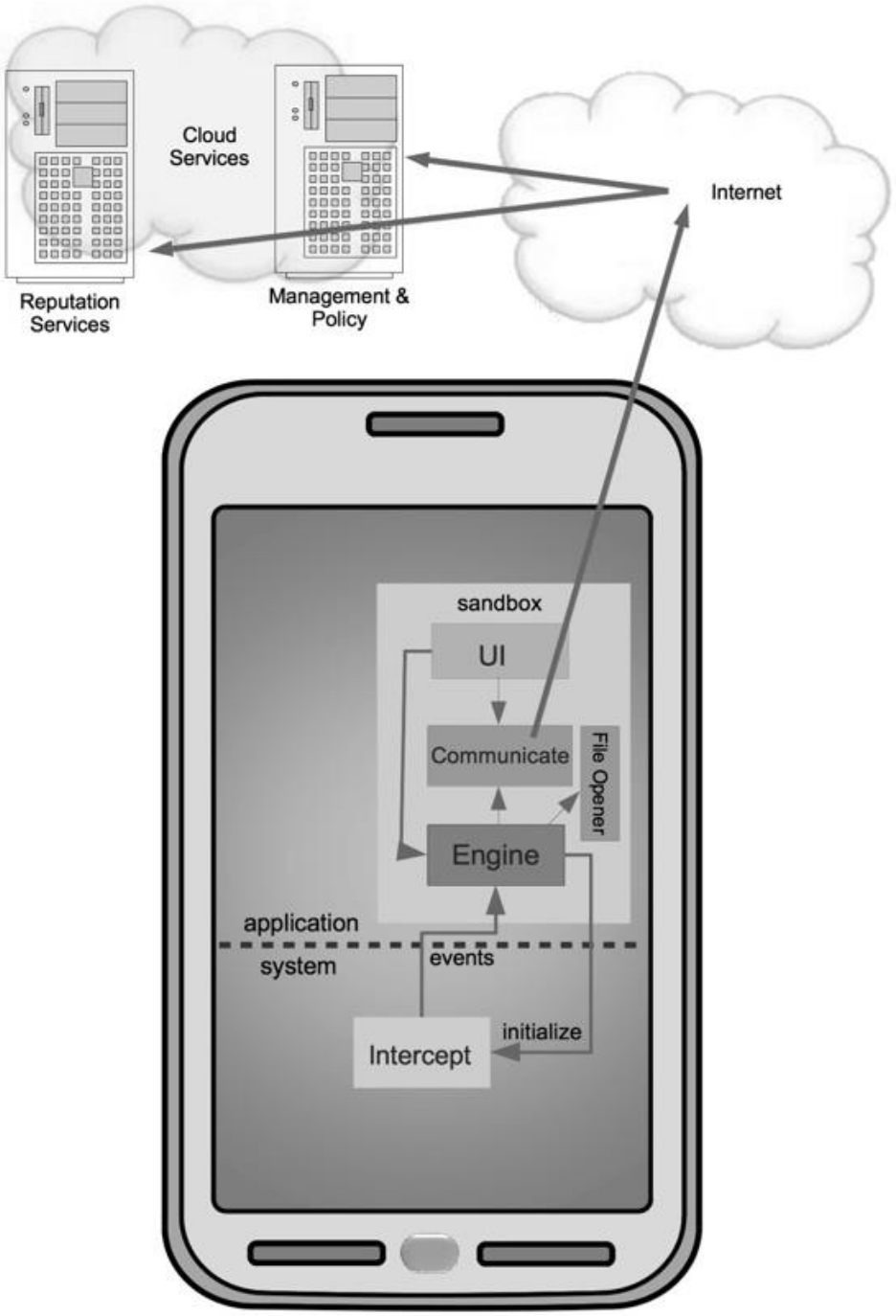
Threat Agent	Goals	Technical Ability	Risk Tolerance	Work Factor	Activity Level
Cybercrime	Monetary	Low (known proven)	Low to medium	Low	Very high, continual
Industrial Espionage	Information	Medium to medium-high	low	medium	Low. For enterprises, medium
Nation-states	Information Disruption	Very high	Very low	Very high	Medium but constant
Law Enforcement/Gov Compliance	Compliance Information	Medium	None – they are the law	medium	intermittent
Insider	monetary	Varies	Low	None	Occasional
Insider	Revenge	Varies	Very high	None	Occasional
Usage abuse	Unauthorized use	Low	Low	Low	constant
Hacktivists	Media attention for cause	Low to medium	Used to be high, now much lower	medium	intermittent
Hackers	Status	Often very low	Low	Low	low
Security Researcher	Career enhancement	High	None	High	medium

Example Analysis

Mobile Security Application

Simple ATASM Steps

- Enumerate CAV
- Define and score impacts
- Enumerate existing mitigations
- Develop requirements to bring system to desired posture
- Prioritize
- Share and review with entire team + product management



[2]

Analysis background and structural points

- Must reside in OS application sandbox
- Intercept in system to grab privileged events of interest for examination

[2] Copyright Brook S.E. Schoenfield, 2015. Used with permission

- Intercept at higher privilege
- Intercept initialized early boot
- Intercept proxies events to engine
- Engine contains decision logic
- UI starts engine & communicate
- Files parsed/normalized by file opener
- External communications proxied through communicate
- Notifications from cloud through OS push notifications
- All message exchanges are initiated from device
- Response to notification
- Device certificate/private key issued at enrollment
- Messages/updates signed by cloud services

Documenting the Model

- The most important document is the requirements output from threat modeling
  - A threat model may be inferred from a thorough security requirements document
  - You must produce a requirements document
- If visual, one or more visual depictions of the architecture
  - Different domains often require different views onto the same system
  - Views might include:
    - attack surfaces
    - Assets
    - Mitigations and controls

Closing Thoughts

Suggestions to Increase Skill

- Find an experienced mentor and 2 peer reviewers
  - Having one reviewer who is Independent from the development team, the architecture the threat model helps to increase the power of the review
- Just do it!
  - Threat models need to be revisited when architecture, threat landscape, and security features change
  - Experienced threat modelers, please make yourself available!
- Start with your own projects
  - Gather the team and wrestle with the threat model
  - To gain experience, you can help with other threat models
    - Modeling diverse architecture deepens abilities

Resources

https://www.owasp.org/index.php/Application_Threat_Modeling
https://www.owasp.org/index.php/Threat_Risk_Modeling
https://www.owasp.org/images/a/aa/AppSecEU2012_PASTA.pdf
http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=427321
http://www.intel.com/Assets/en_US/PDF/whitepaper/wp_IT_Security_RiskAssessment.pdf
https://www.facebook.com/securingsystems
http://www.amazon.com/Securing-Systems-Applied-Security-Architecture/dp/1482233975
https://www.facebook.com/softwaresec
www.amazon.com/Core-Software-Security-Source
http://cybersecurity.ieee.org/images/files/images/pdf/CybersecurityInitiative-online.pdf

brook.e.schoenfield@intel.com (before March 31, 2017)
brook_schoenfield@mcafee.com (after March 31, 2017)
http://www.brookschoenfield.com
brook@brookschoenfield.com
@BrkSchoenfield

RSAConference2017
San Francisco | February 13 – 17 | Moscone Center

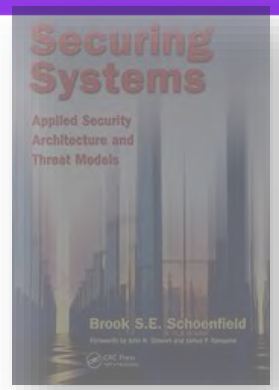
POWER OF OPPORTUNITY

SESSION ID: LAB3-W04

Threat Modeling Demystified

Brook Stephan Schoenfield
Product Architect Product Security
Intel Security
@BrkSchoenfield

Brook S.E. Schoenfield, MBA
Principal Engineer

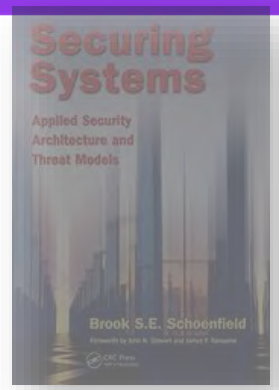


- >1000 security reviews
- 16+ years security
- 30+ years software/hi-tech
- Real-time OS, TCP/IP stack down to hardware
- 5th virtual team lead, 9th technical leader role
- 4th software security lead @ 4th organization

- SANS Institute Featured Speaker, contributor GWEB certification
- Author:
  - *Securing Systems*, CRC Press, 2015
  - *Core Software Security*, CRC Press, 2014
  - *Avoiding The Top 10 Security Design Flaws, IEEE*
  - Various SANS Smart Guides, papers, etc.
- Founding member IEEE Center for Secure Design
- Featured security architect at Bletchley Park Cyber Museum

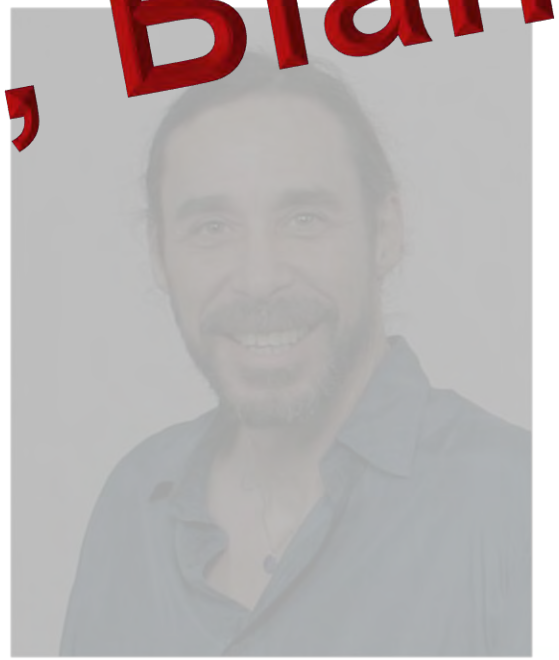
intel Security

Brook S.E. Schoenfield, MBA
Principal Engineer



- >1000 security reviews
- 16+ years security
- 30+ years software/hi-tech
- Real-time OS, TCP/IP stack down to hardware
- 5th virtual team lead, 9th technical leader role
- 4th software security lead @ 4th organization

Blah, Blah, Blah

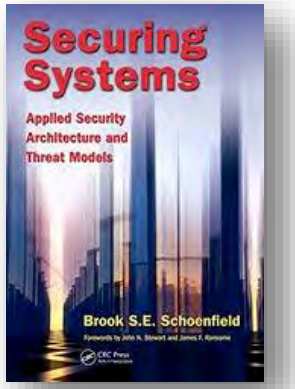


- SANS Institute Featured Speaker, contributor GWEB certification
- Author
  - *Securing Systems*, CRC Press, 2015
  - *Core Software Security*, CRC Press, 2014
    - *Avoiding The Top 10 Security Design Flaws, IEEE*
    - Various SANS Smart Guides, papers, etc.
- Founding member IEEE Center for Secure Design
- Featured security architect at Bletchley Park Cyber Museum

intel Security

First Comments

- ATASM[1] is a pedagogy
- ATASM is a mnemonic, a high level abstraction for a process that is often non-linear, and highly recursive
- “Peel the onion” is close to reality, but harder to learn and to coach: a real threat model process is often quite fractal
- You are encouraged to use ATASM; if useful, please teach

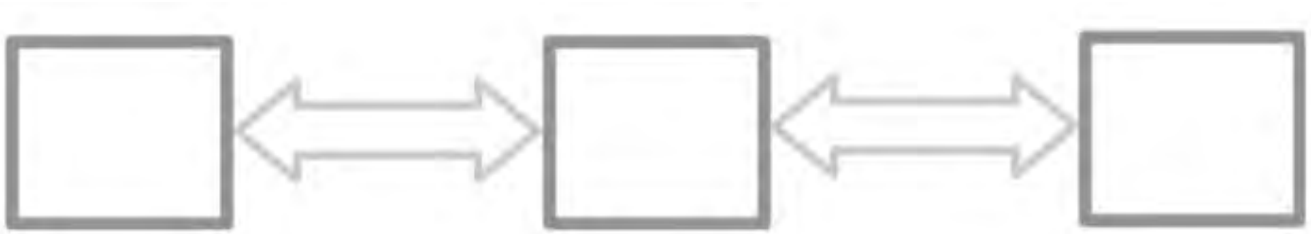


1. ATASM/risk material © Brook S.E. Schoenfield, 2010-2015, all rights reserved, CRC Press, 2015. Used with permission

Possible Outcomes From This Session

- Newbies: Introduction to threat modeling concepts
- Learners: Dig deeper, gain consistency, increase comfort, practice
- Practitioners: Conceptualization and explanation; a method that fosters inclusion
- Experts: Articulate in order to present and to teach
- Everyone: Hopefully, some new tricks of the trade?

What Can Be Said About This “Architecture”?



Is This Any Better?

Is there a recurring design problem: Have you heard about or experienced poor security design choices?

Break into work groups (4-6) to brainstorm examples of insecure design.

Design brainstorm report back from workgroups

Consider Recent Design Misses

That Jeep Wrangler

Open WiFi Pacemaker

The Target breach was a system design failure

IEEE CENTER FOR SECURE DESIGN

A start…

http://cybersecurity.ieee.org/images/files/images/pdf/CybersecurityInitiative-online.pdf

Threat modeling is a technique to identify the attacks a system[1] must resist and the defenses that will bring the system to a desired defensive state

1. “system” is defined inclusively

The Timing Component: Earlier is Better

Early requirements get the "worm"

Acknowledgement and Planning

Or at least, early requirements capture team mindshare

Start the threat model at the point when the architecture is sufficiently defined such that most if not all of the major functional components are understood

Revisit the threat model when the architecture materially changes.

Threat models are living documents.

Threat modeling is compatible with iterative development practices. Iterate the threat model appropriately.

Tool Belt

Architecture is a tool for structuring complexity

Architecture provides a playground for potential changes

Architecture Risk Assessment (ARA) is a tool for applying a threat landscape to an architecture to uncover security needs

---

Threat modeling is a tool for secure design.

A threat model is not a “design”

Threat model is dependent upon architectural understanding

Threat models output security requirements

Everything You Know About Security…

Architecture is a tool for structuring complexity

Architecture is a tool for playing with potential changes

Architecture Risk Assessment (ARA) is a tool for applying a threat landscape to an architecture to discover security needs

Threat modeling is applied security architecture

Threat modeling is a tool for secure design.

A threat model is not a “design”

Threat model is dependent upon architectural understanding

Threat models output security requirements

“Secure” Software Must:

- Be rid of implementation errors with exploitable effects
- Contain customers’ security features
- Be self-protective
- Minimize consequences of successful attack (“fail well”)
- Install with sensible, “closed” defaults

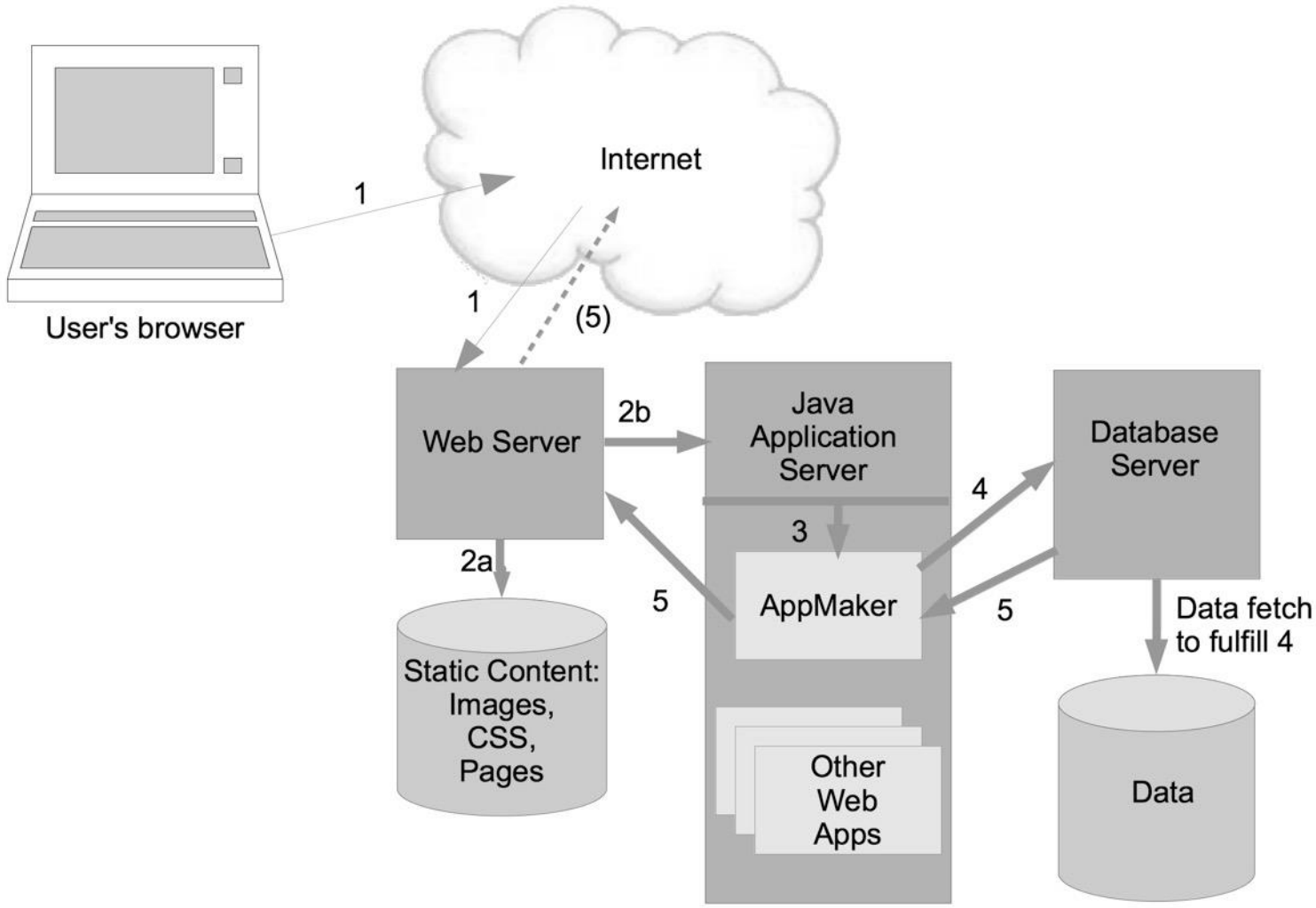
Design Secure Software

- Be rid of implementation errors with exploitable effects

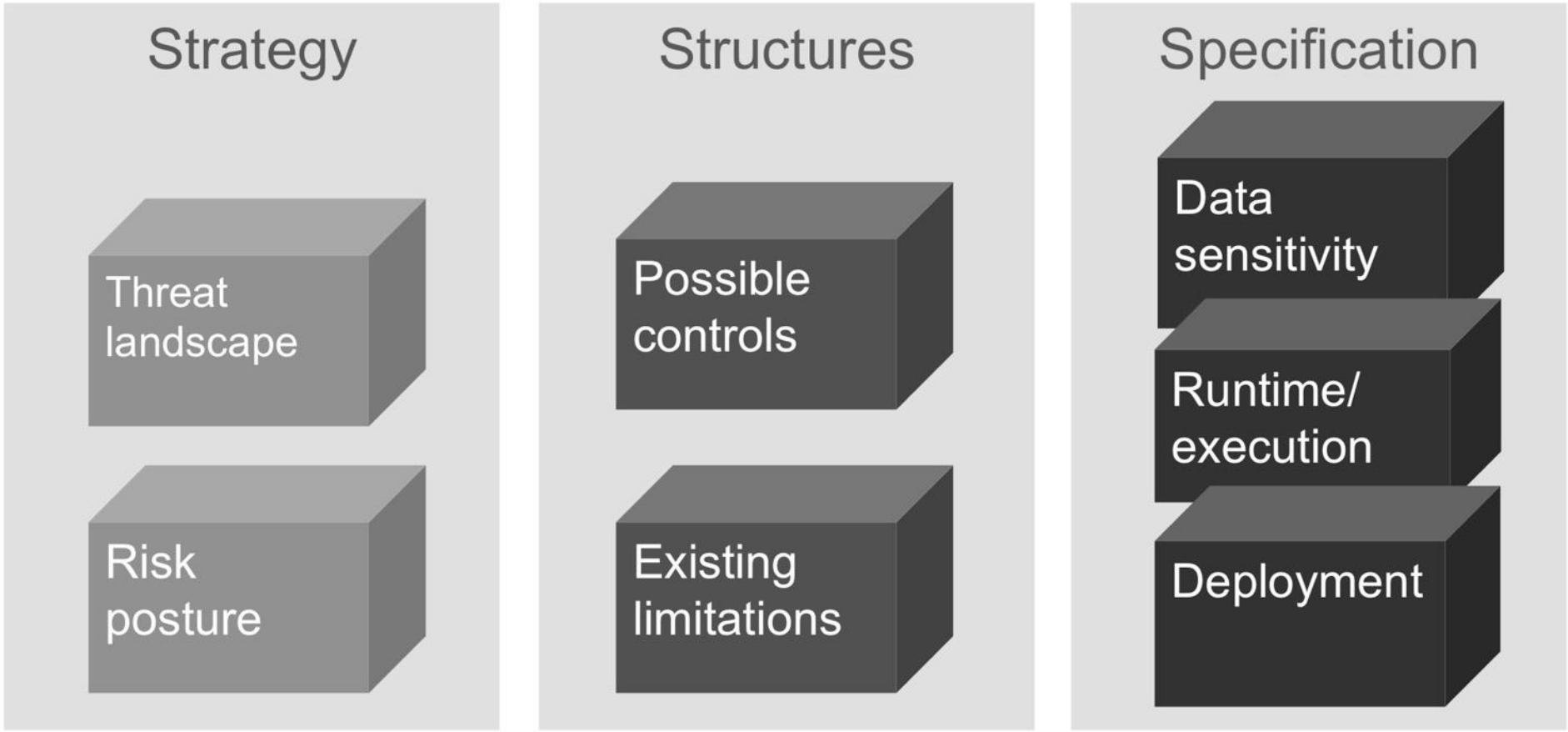
- Contain customers' security features
- Be self-protective
- Minimize consequences of successful attack ("fail well")
- Install with sensible, "closed" defaults

ARA & threat modeling help with these tasks

What Can We Infer?
What More Do We Need And Why?

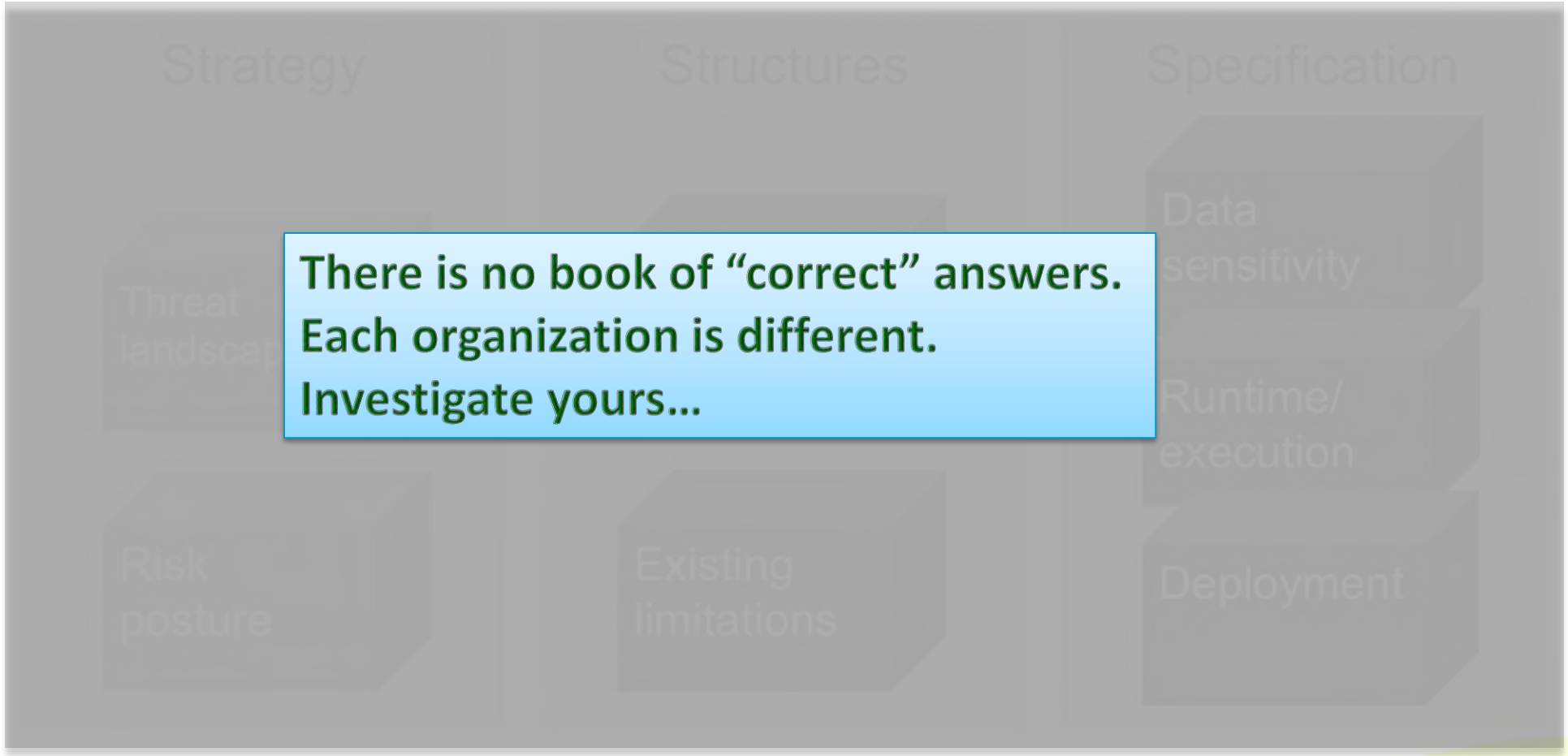


Prework: The 3 S's



Securing Systems, CRC Press, 2015

Background Information Is Critical



Securing Systems, CRC Press, 2015

A threat model is a crossroads of knowledge from architecture experts, domain experts, and security experts

Absence of one or more stakeholders cripples the model and its usefulness

Personal accelerometer

- Measures movement of body
- Worn on wrist
- Extremely low power
- Long battery life
- Minimal operating system and CPU

Low Power Bluetooth Specification

- There is no protection of communications!
  - E.g., TLS not used over open air
- Authentication by device ID only
- Devices only exchange movement data

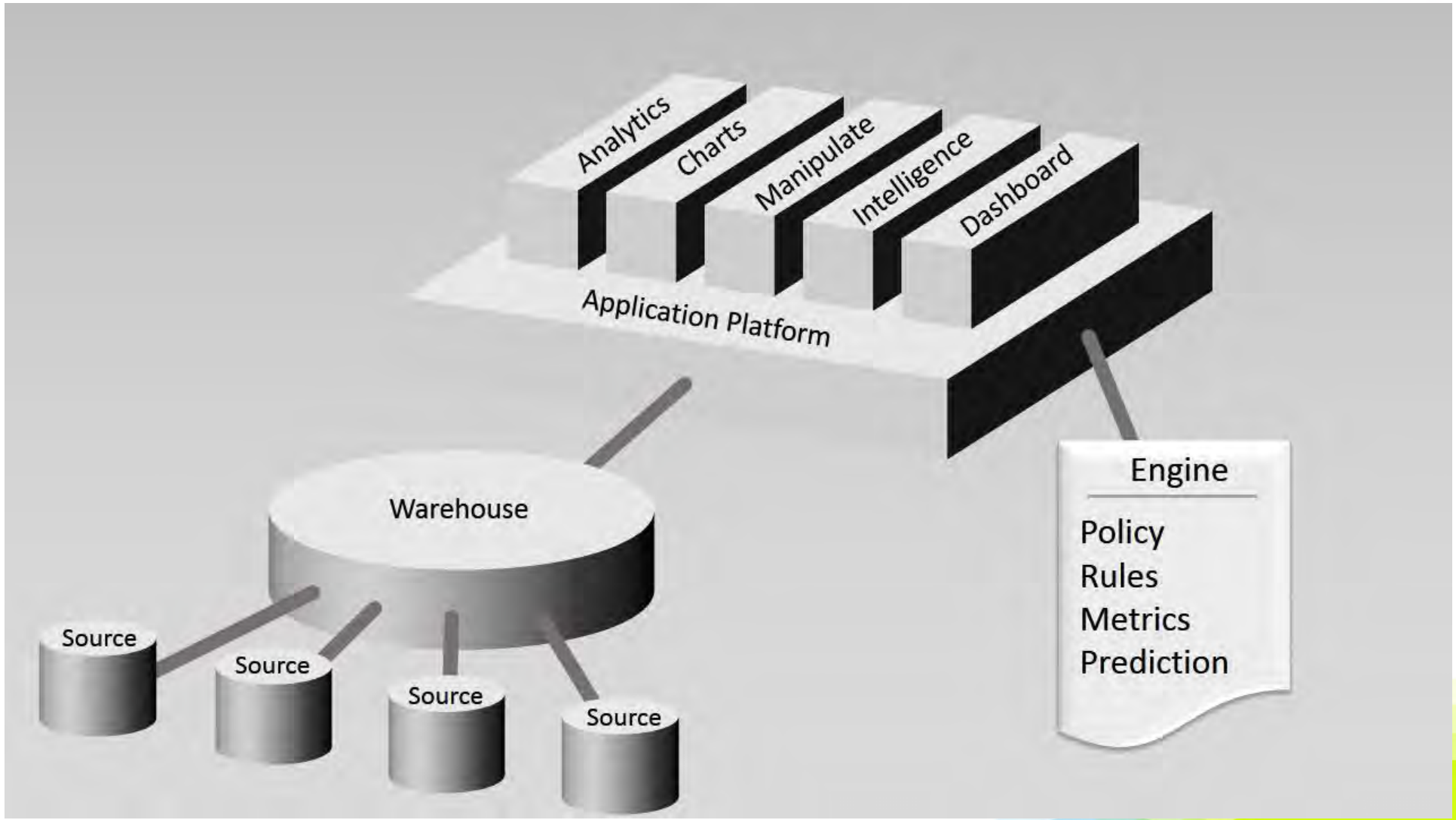
Huh?

Understanding system architecture is key

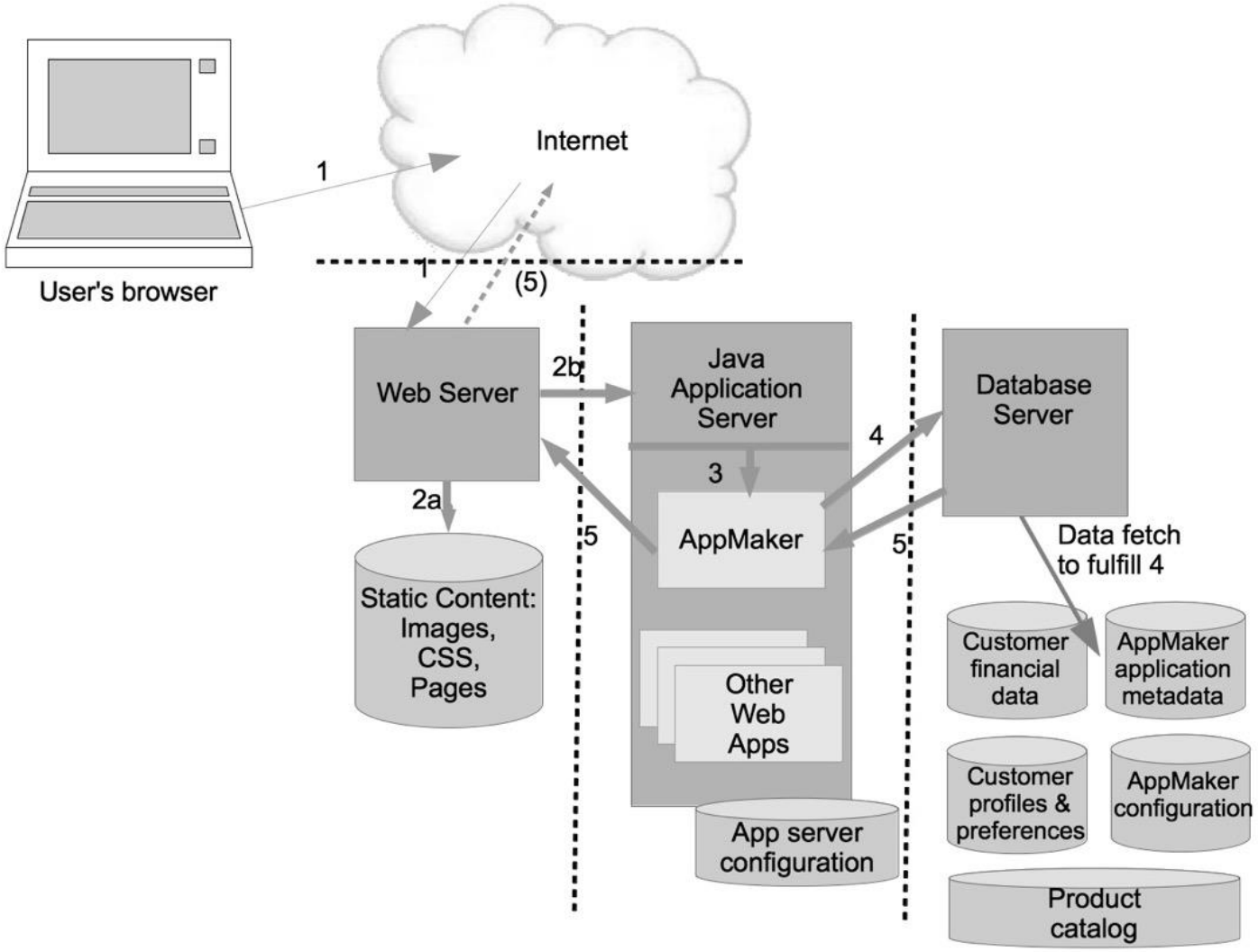
Differing views and levels of detail each solve distinct problems

There is no perfect view or level

A Threat Model From This?



Web-Sock-A-Rama: Any More Information?



Web-Sock-A-Rama Web Business

- Successful Online Sock store
- All kinds of socks: men, women, children, specialty, sports, artistic, novelty
- Takes no public political positions
- Relatively risk averse
- Customers are considered the business' greatest resource
  - Customer security and privacy prime objectives
- Assume that the support and administration of the infrastructure is run rigorously:
  - strong access controls and need-to-know (e.g., NIST-800-53): networks, hosts, execution environments
  - External network separated from internal
  - Administration is via highly controlled management network

Start At High Level

What is the organization's purpose?

What is this system's purpose?

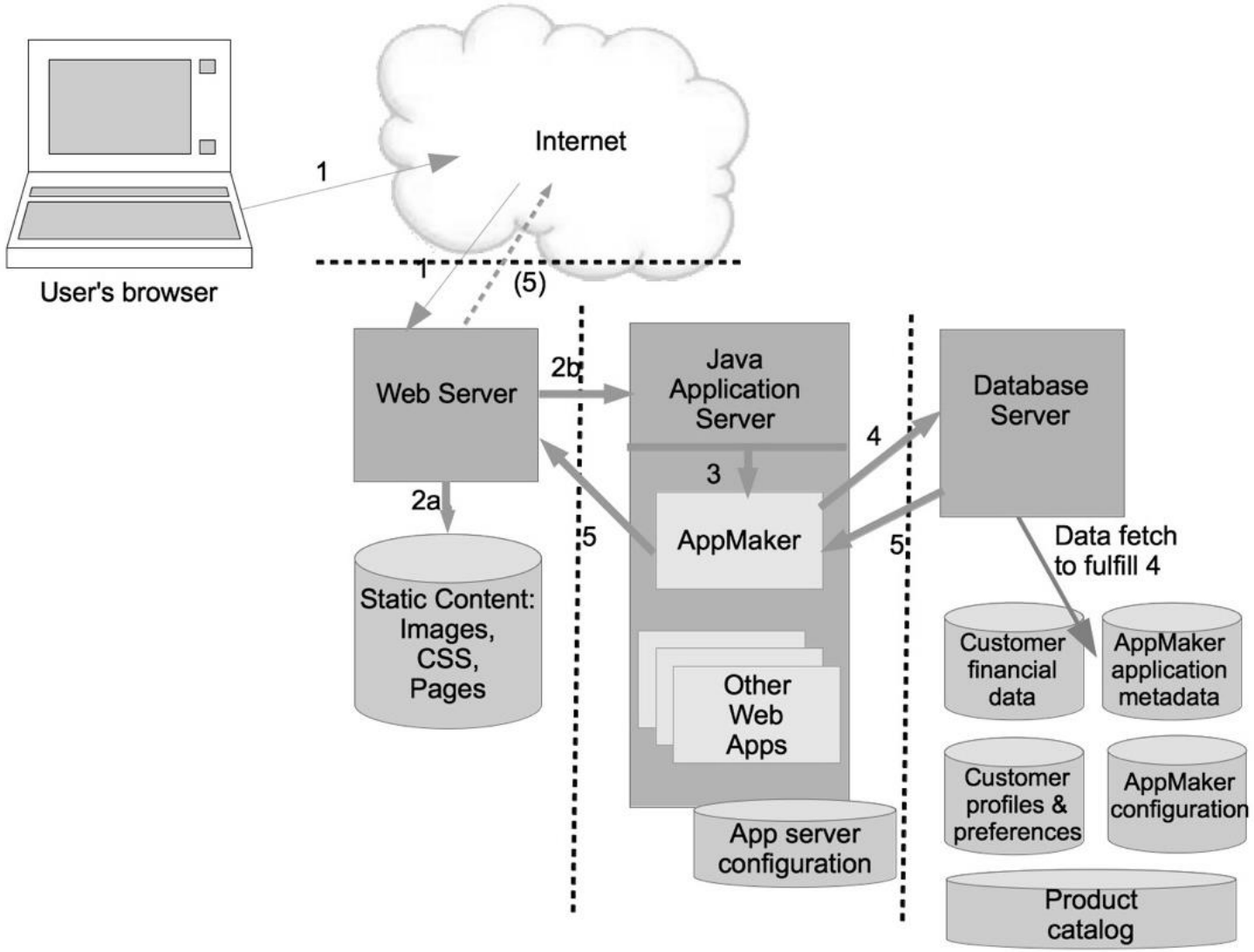
How does this system contribute to the organization goals? To an enterprise architecture?

What are the major functions?

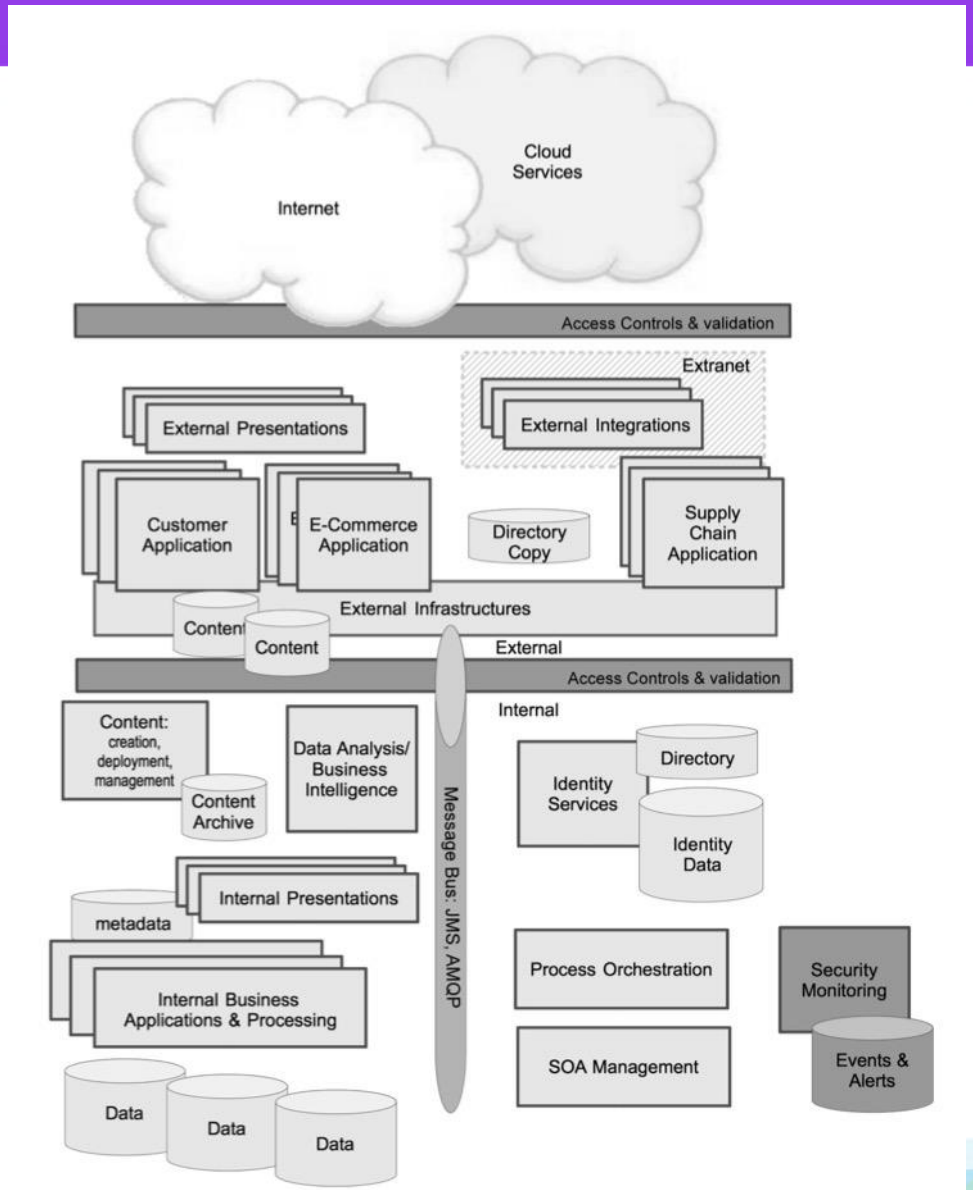
How do these interact? Why?

Derive, if you can, the intended risk posture

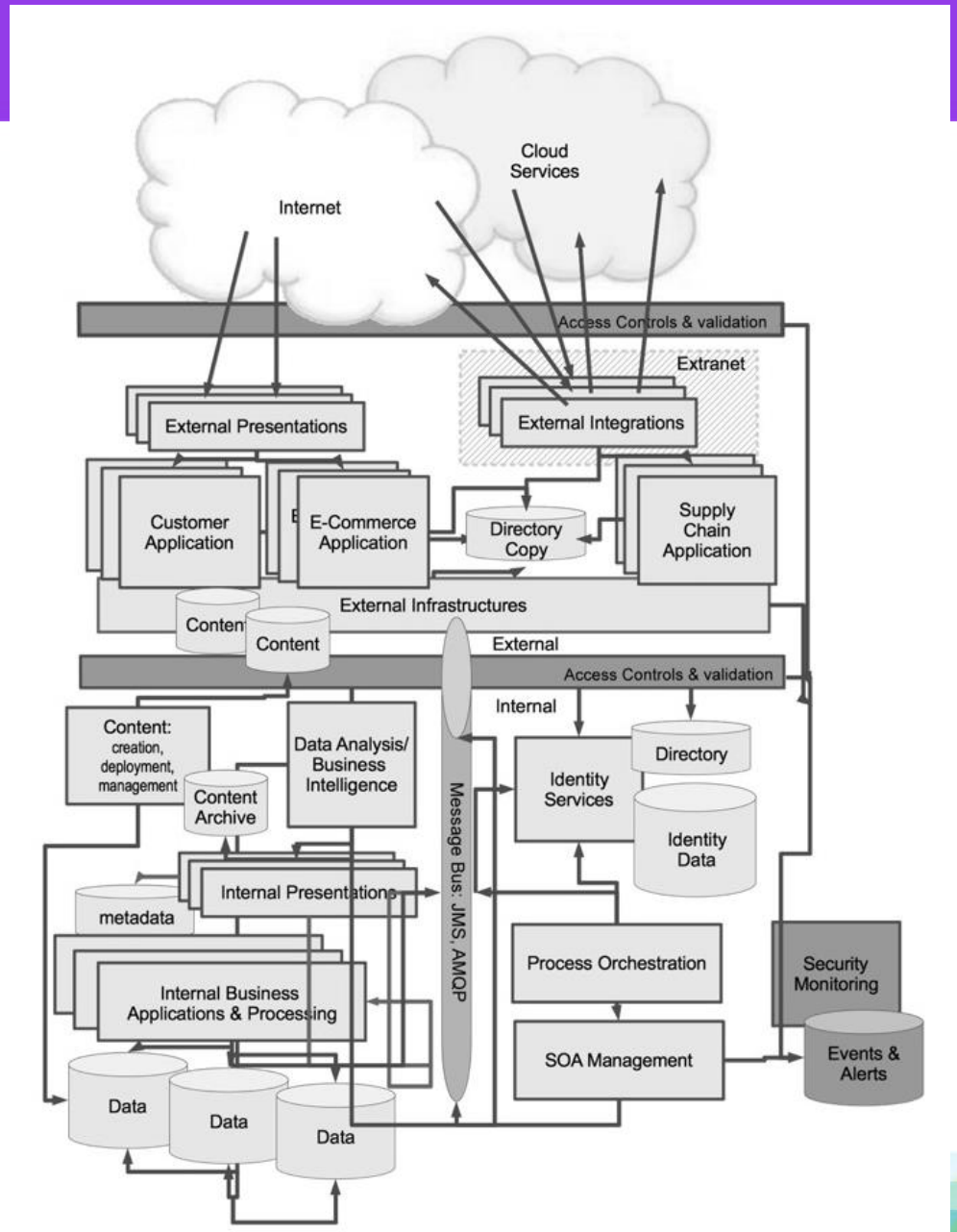
Web-Sock-A-Rama
Find an attack vector



Logical Or Component Level



Useful View?


Cloud Services
Internet
Access Controls & validation
Extranet
External Presentations
External Integrations
Customer Application
E-Commerce Application
Directory Copy
Supply Chain Application
External Infrastructures
Content
Content
External
Access Controls & validation
Internal
Content: creation, deployment, management
Content Archive
Data Analysis/ Business Intelligence
Message Bus: JMS, AMQP
Identity Services
Directory
Identity Data
Internal Presentations
metadata
Process Orchestration
Security Monitoring
Internal Business Applications & Processing
SOA Management
Events & Alerts
Data
Data
Data

ATASM

Threat Modeling = Applied security architecture

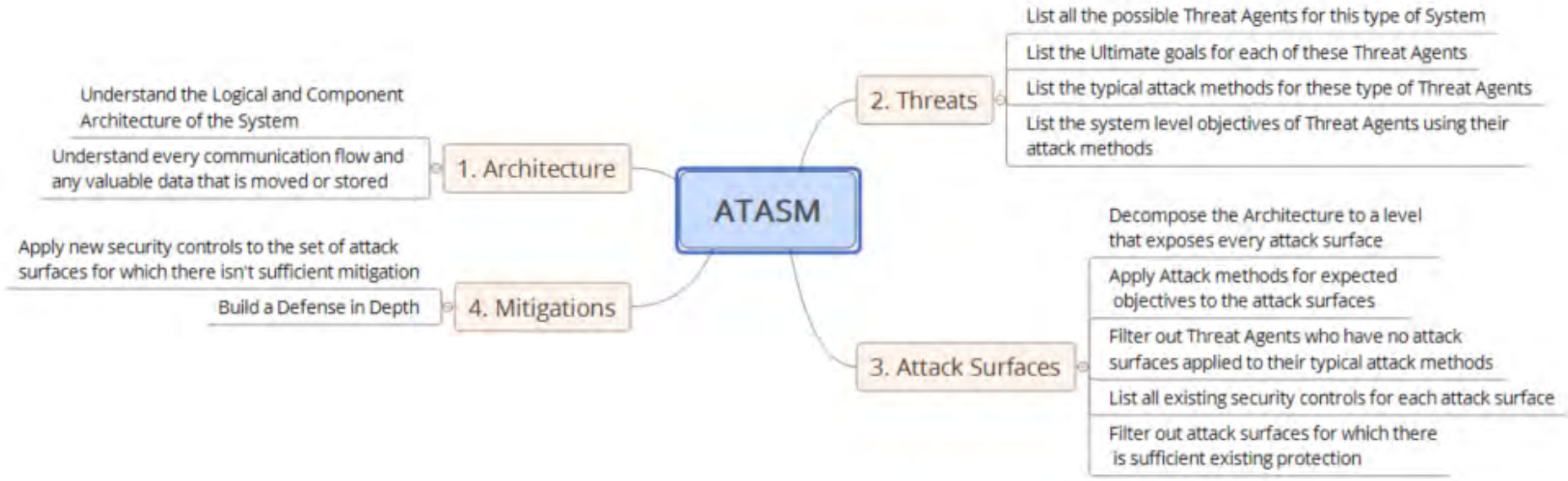
Securing Systems, CRC Press, 2015

ISecG ESPT Security Test Plan Report [Product]

Substeps

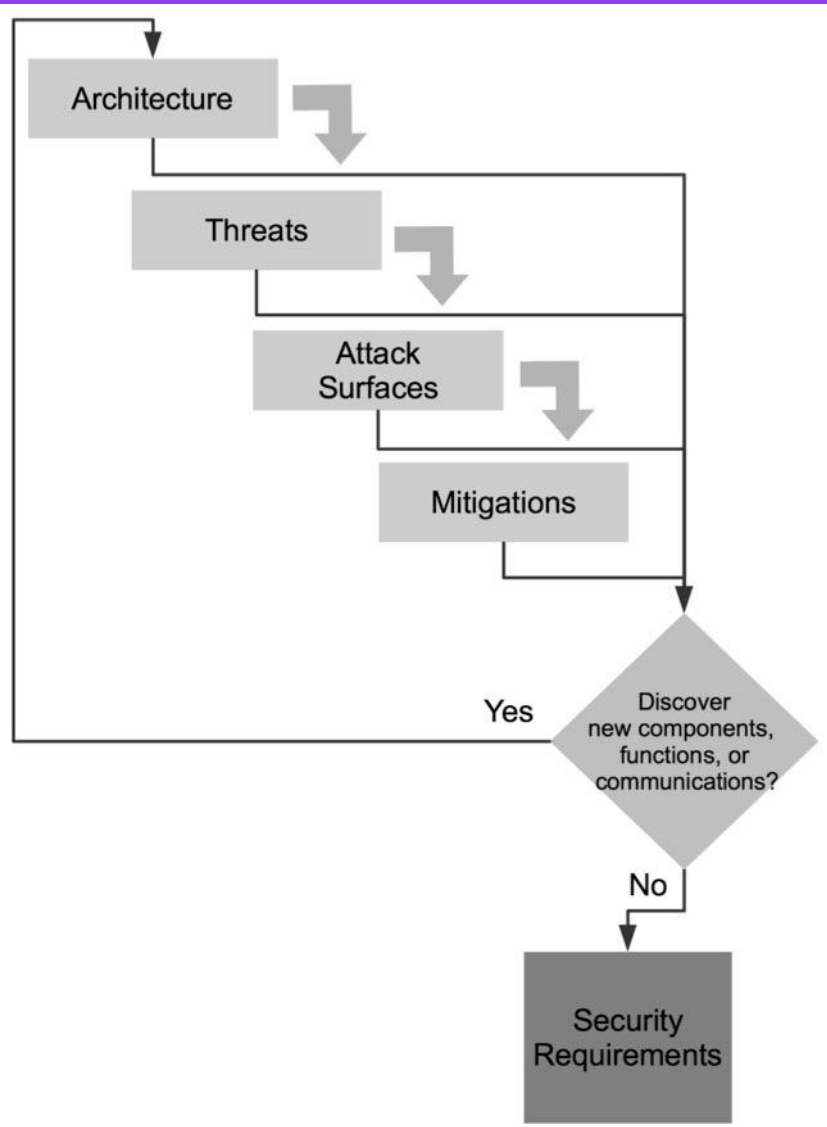
Architecture/Design Review

ATAM

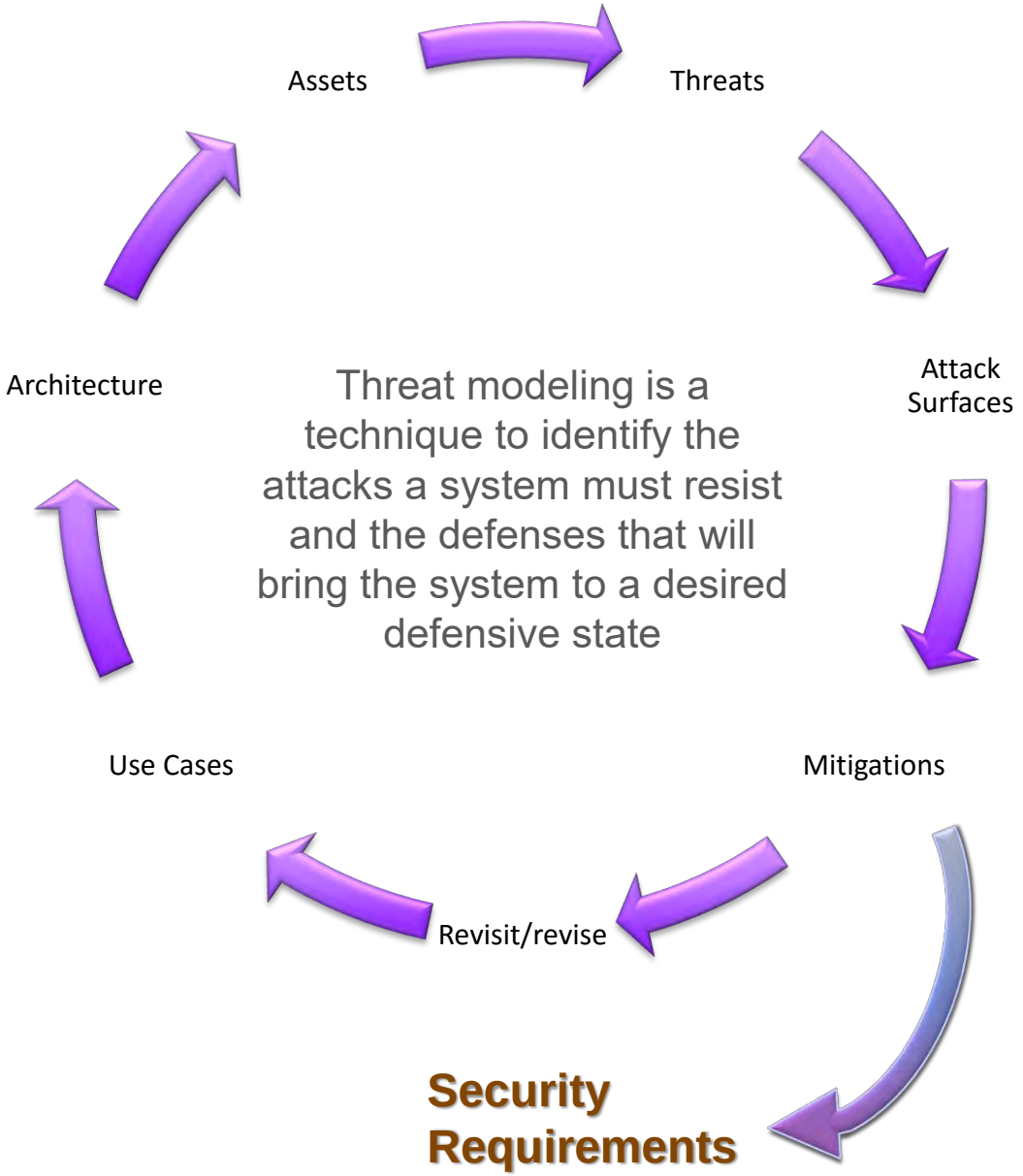


Securing Systems: Applied Security Architecture and Threat Models, *Brook S.E Schoenfield* [2015]

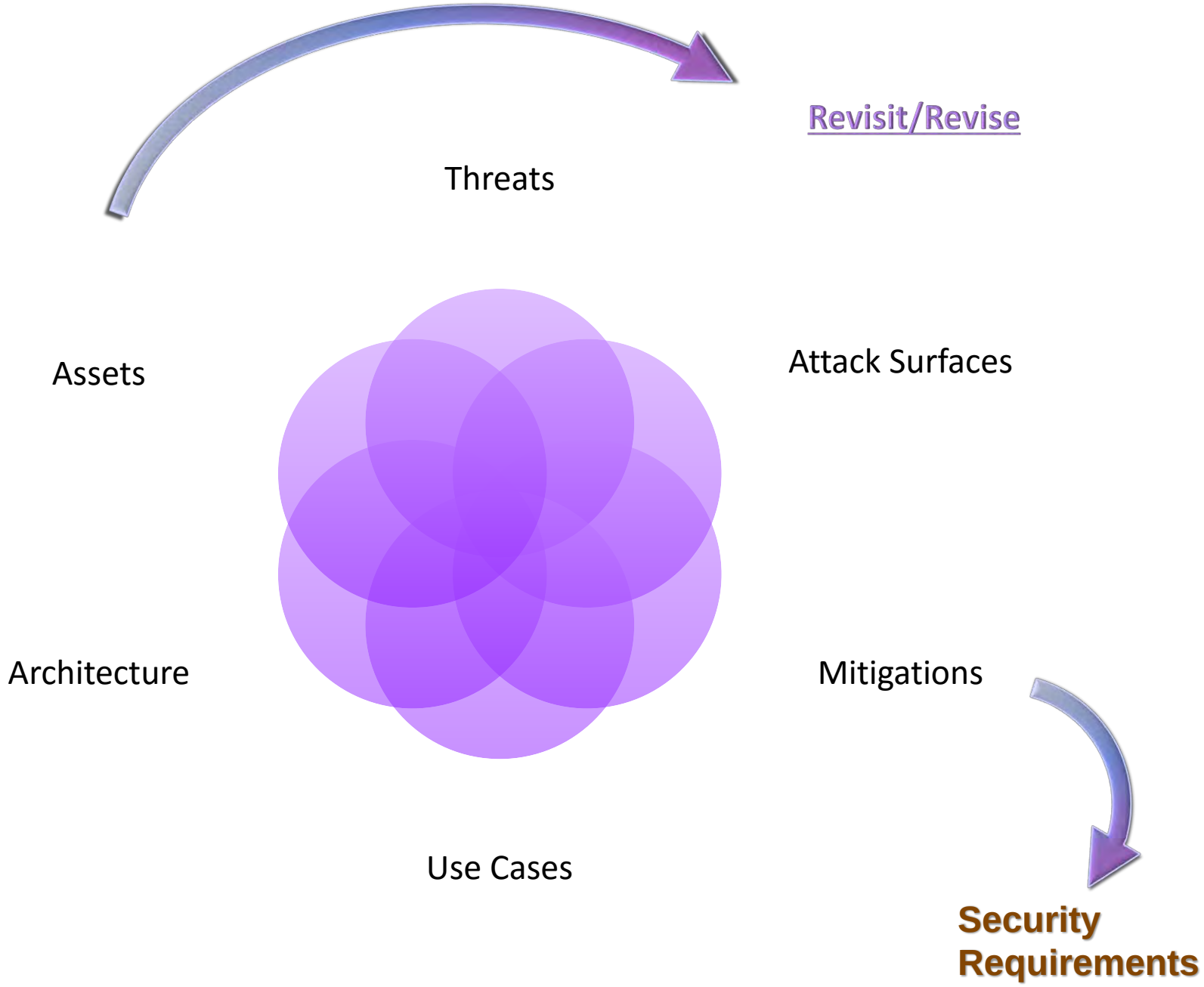
Fractal → Recursive



Securing System, CRC Press, 2015



By Sung Lee; concept by David Wheeler; used with permission



Adapted from original by Sung Lee; concept by David Wheeler; used with permission

The outcome, the ”point” is to discover the necessary security requirements

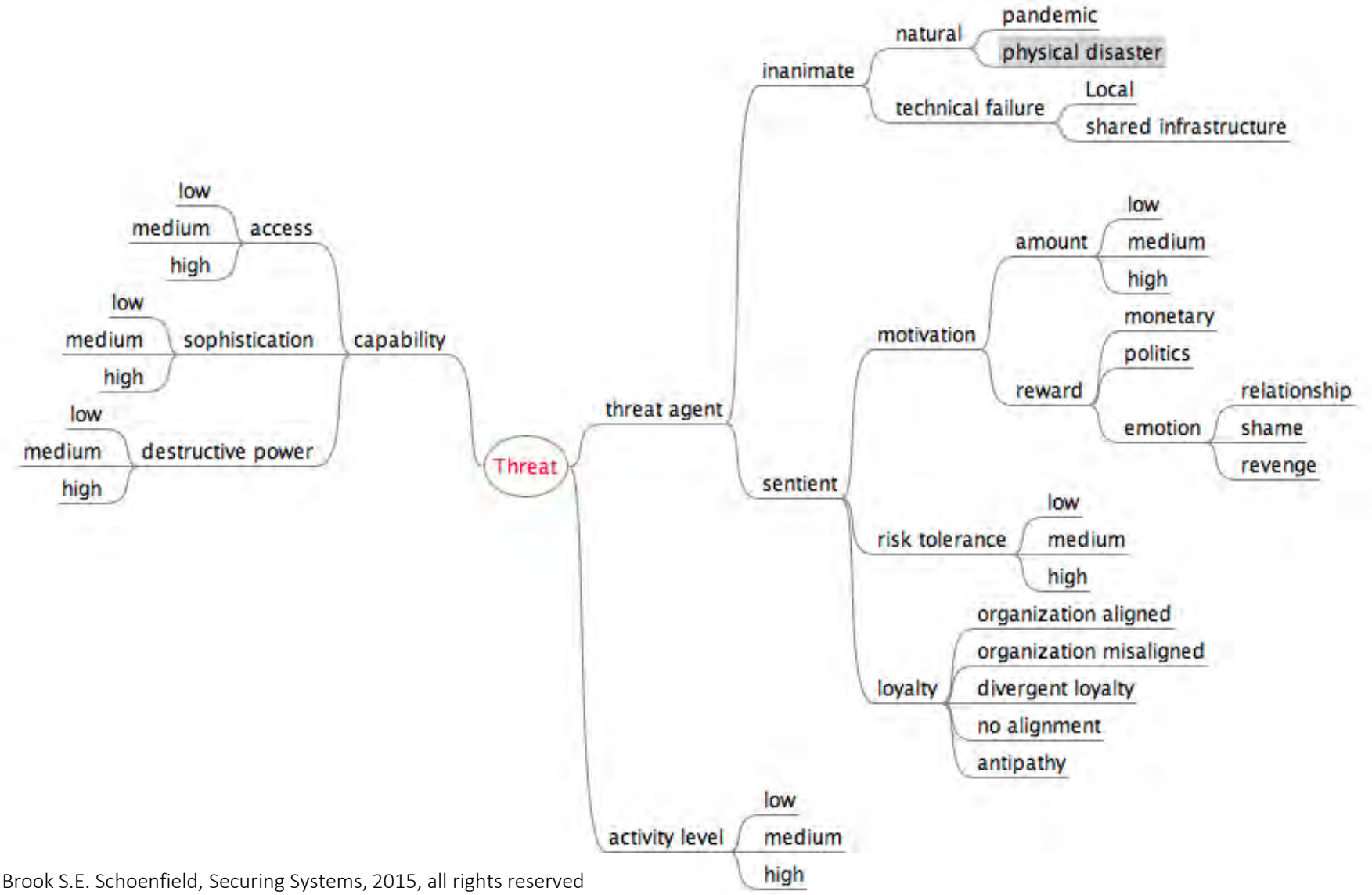
“Security needs tend to get expressed as system constraints” – Erik Simmons

Digging into threat agents and their methods

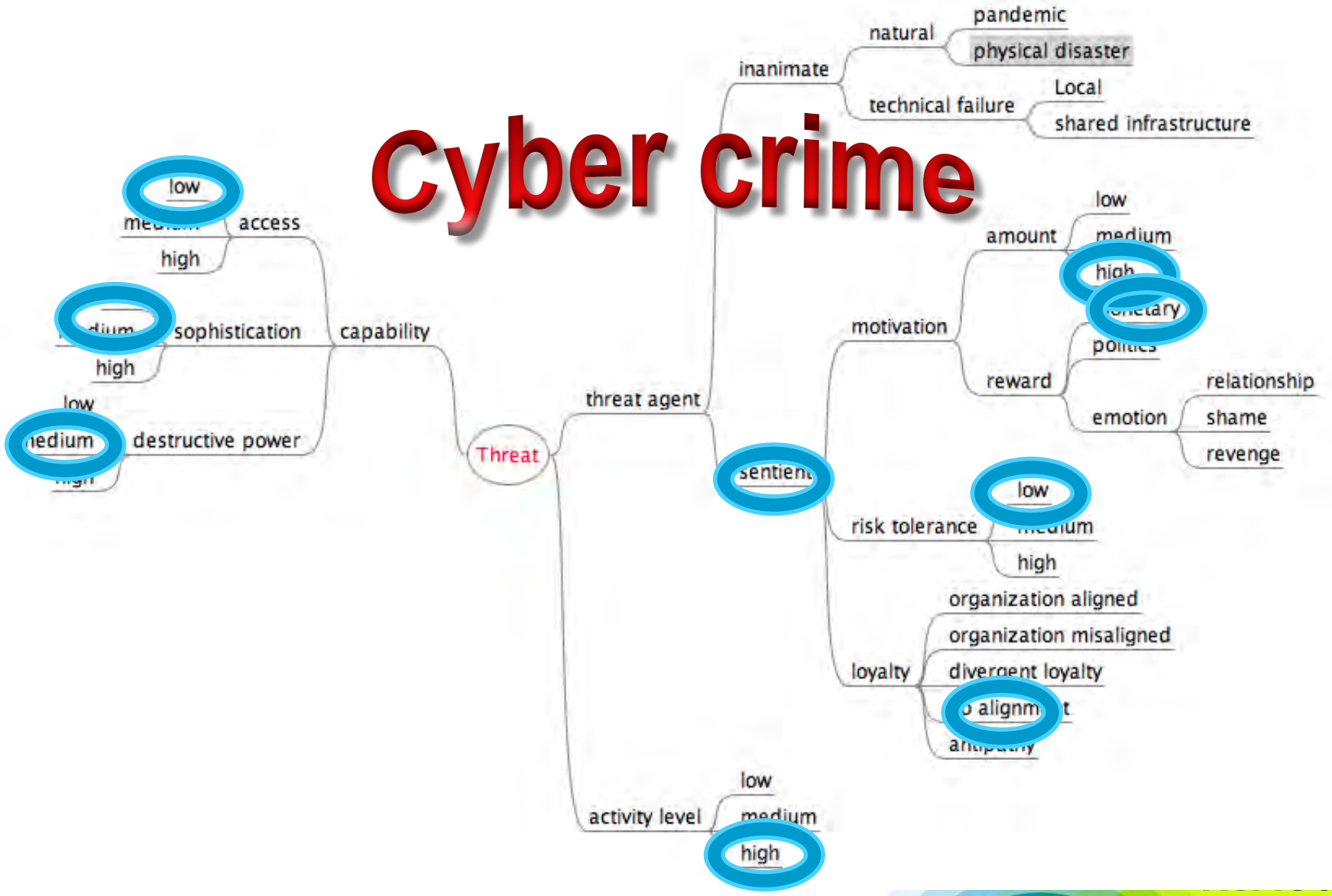
Focus on human actors

Adversaries are adaptable and creative

Threat Attributes



Cyber crime


Threat
capability
access
low
high
sophistication
high
destructive power
low
threat agent
inanimate
natural
pandemic
physical disaster
technical failure
Local
shared infrastructure
sentient
motivation
amount
low
medium
high
reward
relationship
shame
revenge
risk tolerance
low
high
loyalty
organization aligned
organization misaligned
divergent loyalty
activity level
low
medium
high

Threat Landscape

Table 2.1 Summarized Threat Attributes

Threat Agent	Goals	Risk Tolerance	Work Factor	Methods
Cyber criminals	Financial	Low	Low to medium	Known proven
Industrial spies	Information and disruption	Low	High to extreme	Sophisticated and unique
Hacktivists	Information, disruption, and media attention	Medium to high	Low to medium	System administration errors and social engineering

- Divide paper into 5 columns

Threat Agent, Goals, Technical ability (Methods), Risk tolerance, Work Factor

- Start by asking, “Who will want to attack this piece of software, when it runs under its normal, expected deployment?”
- No discussion about the merits of an idea. (brainstorm principle)
- Collate brainstorm ideas into the larger buckets. Be stereotypical
- Derive your group’s threat agent landscape

Table 2.1 Summarized Threat Attributes

Threat Agent	Goals	Risk Tolerance	Work Factor	Methods
Cyber criminals	Financial	Low	Low to medium	Known proven

- Divide paper into 5 columns

Threat Agent, Goals, Technical ability (Methods), Risk tolerance, Work Factor

- Start by asking, “Who will want to attack this piece of software, when it runs under its normal, expected deployment?”
- No discussion about the merits of an idea. (brainstorm principle)
- Collate brainstorm ideas into the larger buckets. Be stereotypical
- Derive your group’s threat agent landscape

Put our matrices together

Set priority with risk

Risk != vulnerability

Threat != vulnerability

Threat != exploit

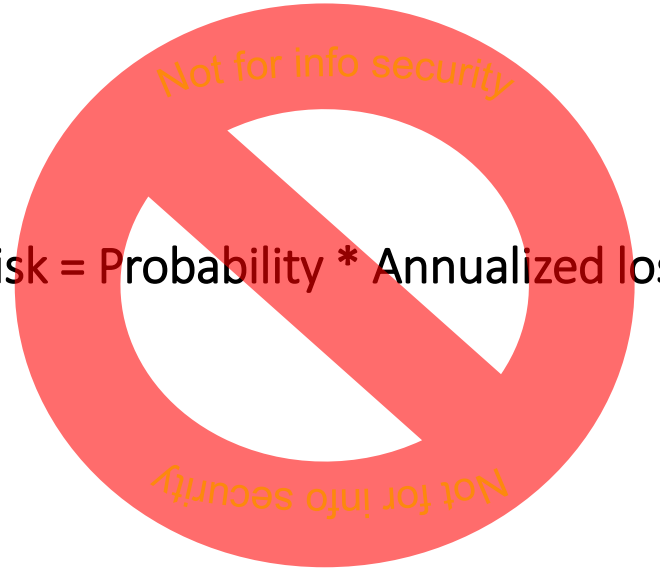
Set Priority With Risk

- A usable risk rating method *should* be:
  - Lightweight
  - Fast
  - Simple and understandable
  - Intuitive

Risk = Probability * Annualized loss

Set Priority With Risk

- A usable risk rating method *should* be:
  - Lightweight
  - Fast
  - Simple and understandable
  - Intuitive



We haven't actuarial tables

Successful attack requires an active threat agent who has the motivation to attack that particular system and who uses the threat agent's skills (exploit) via an exposed (attack surface) exploitable weakness (vulnerability) that can be manipulated by that attacker.

Threat + exploit + exposure + vulnerability

The “kill chain”

For Threat Models Only!

- "Credible Attack Vector" (CAV)
- CAV == Threat & Exploit & Exposure & Vulnerability

  (consider each term as essentially boolean for this narrow context)

Interrupt any CAV term and you interrupt the kill chain

E.g., Threat & Exploit & !~~Exposure~~ & Vulnerability != CAV

(break the kill chain; eliminate the CAV; lower risk exposure)

- Given 0 < CAV < 1, then risk might be expressed as:
  - **Risk Rating = CAV * Impact**

    (essentially how JGERR rates risks)

Now That You Know, Let's Have Some Fun…

- Threats ↴
- Architecture ↴
- Attack Surfaces ↴
- Mitigations

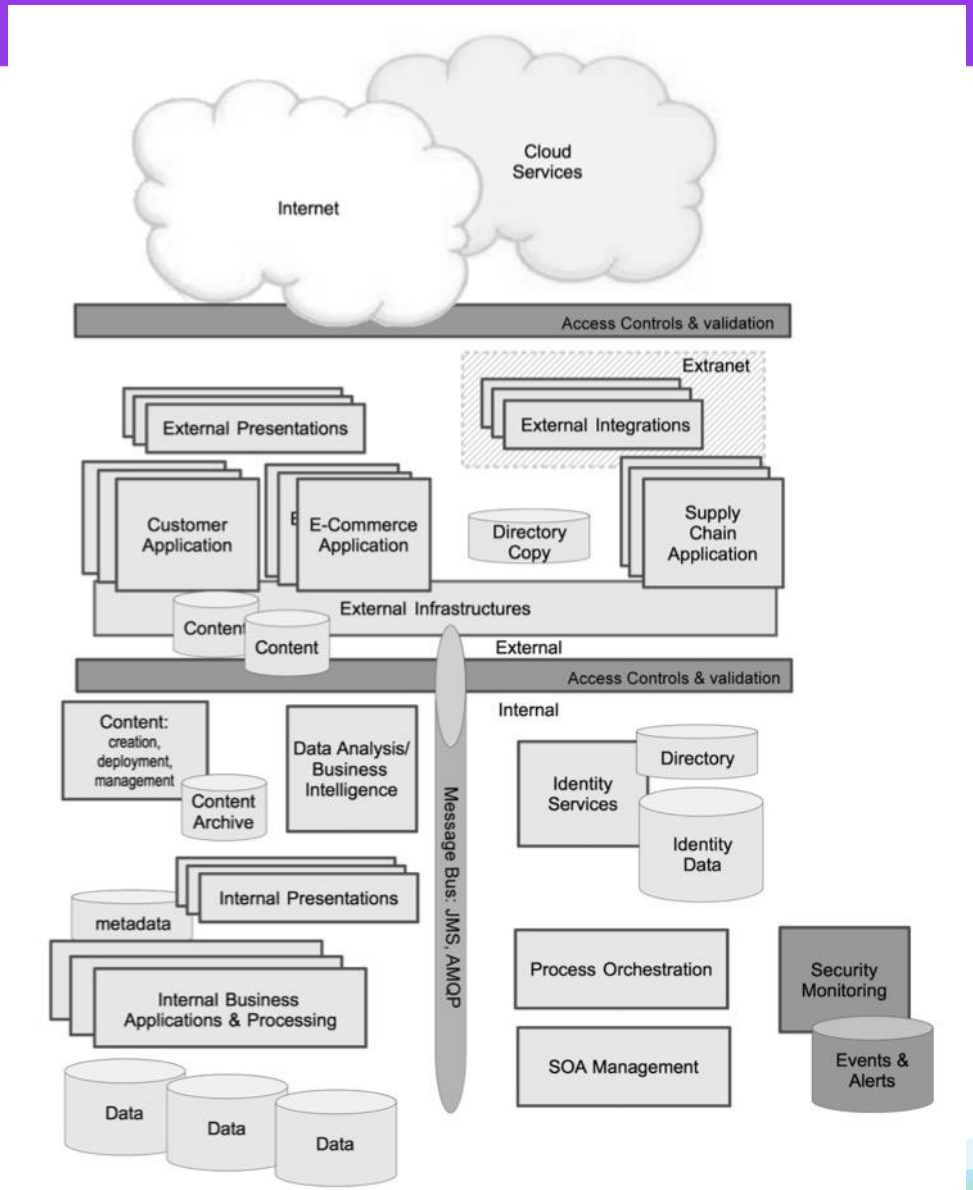
our goal is to experience each step, not to finish the threat model

Yes, I switched the order for teaching purposes

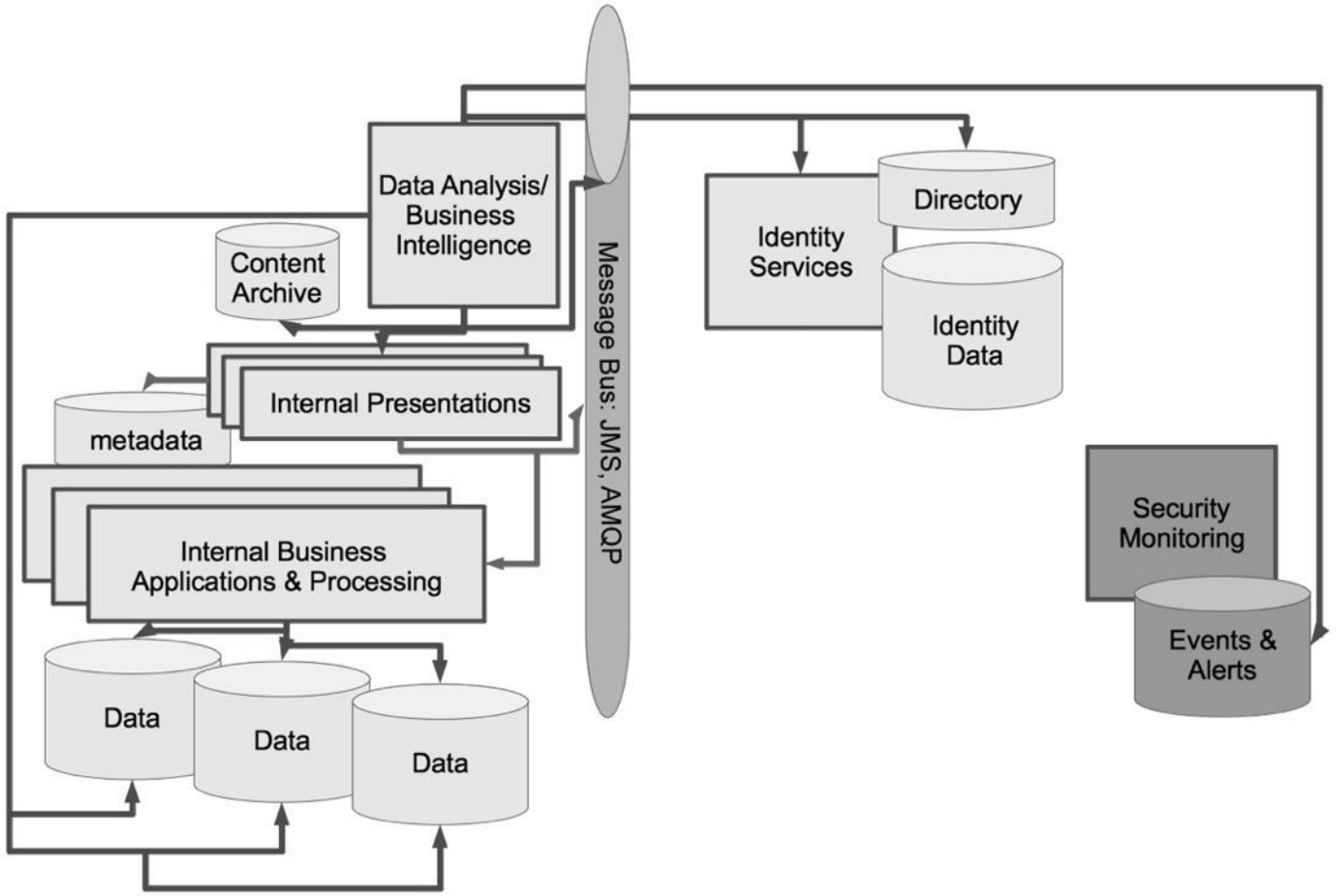
Threat modeling Examples

Fictitious but based on real systems

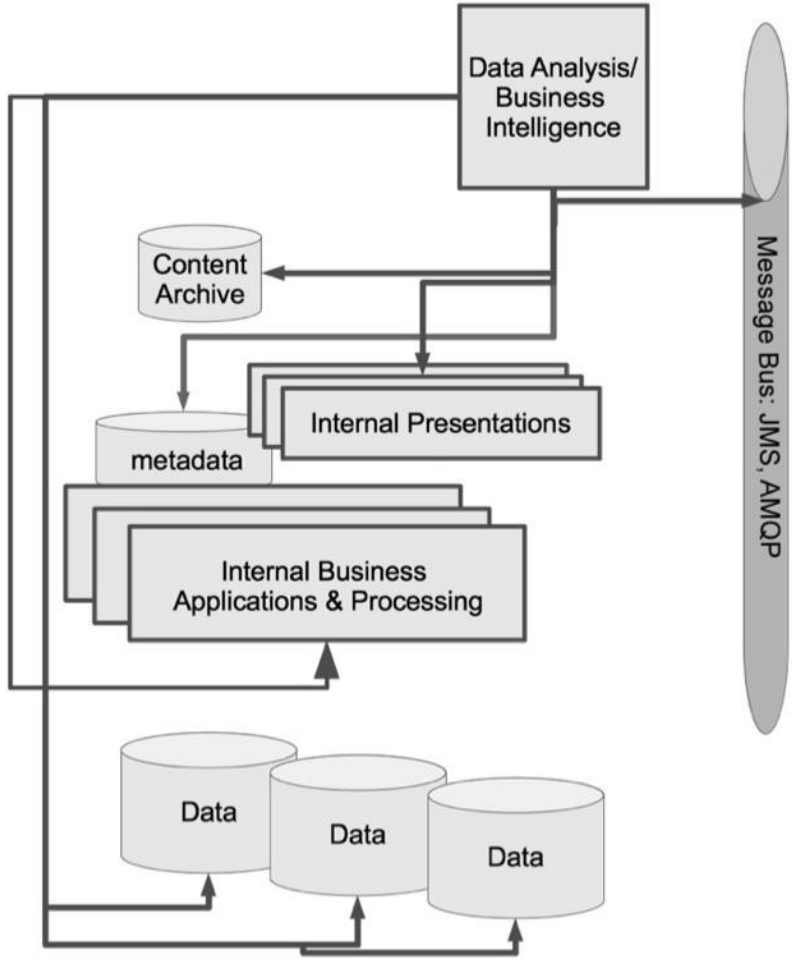
Logical Or Component Level



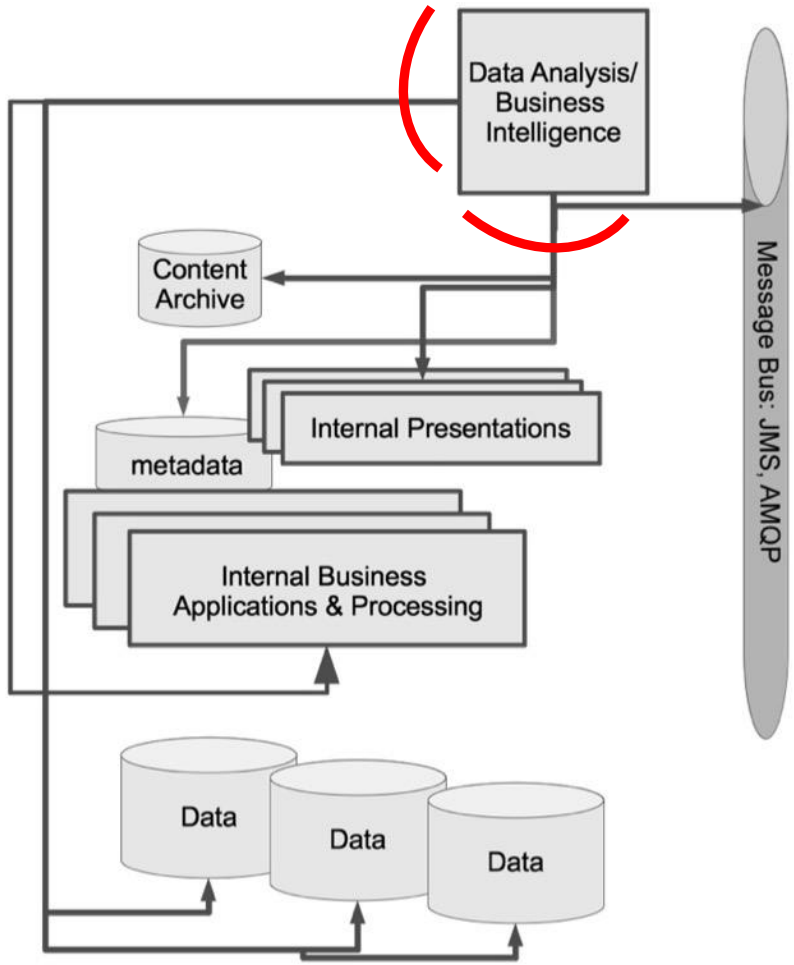
Business Intelligence (Analytics) @Digital Diskus



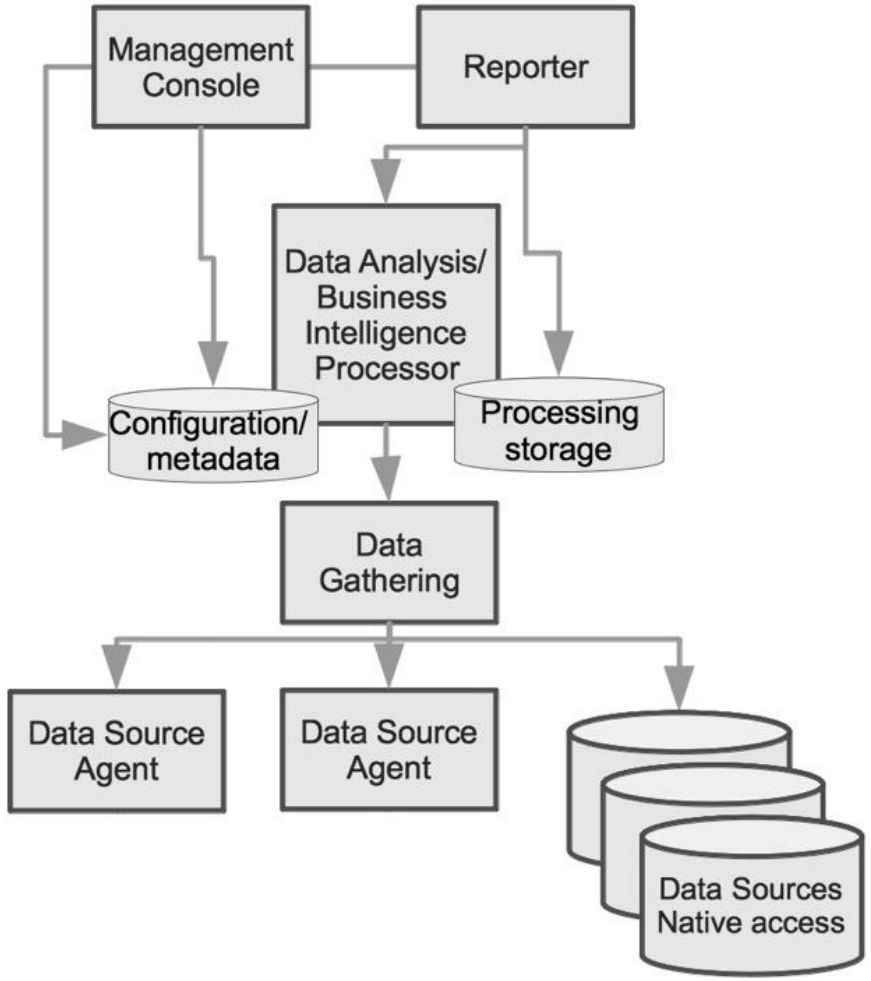
Analytics Data Harvesting



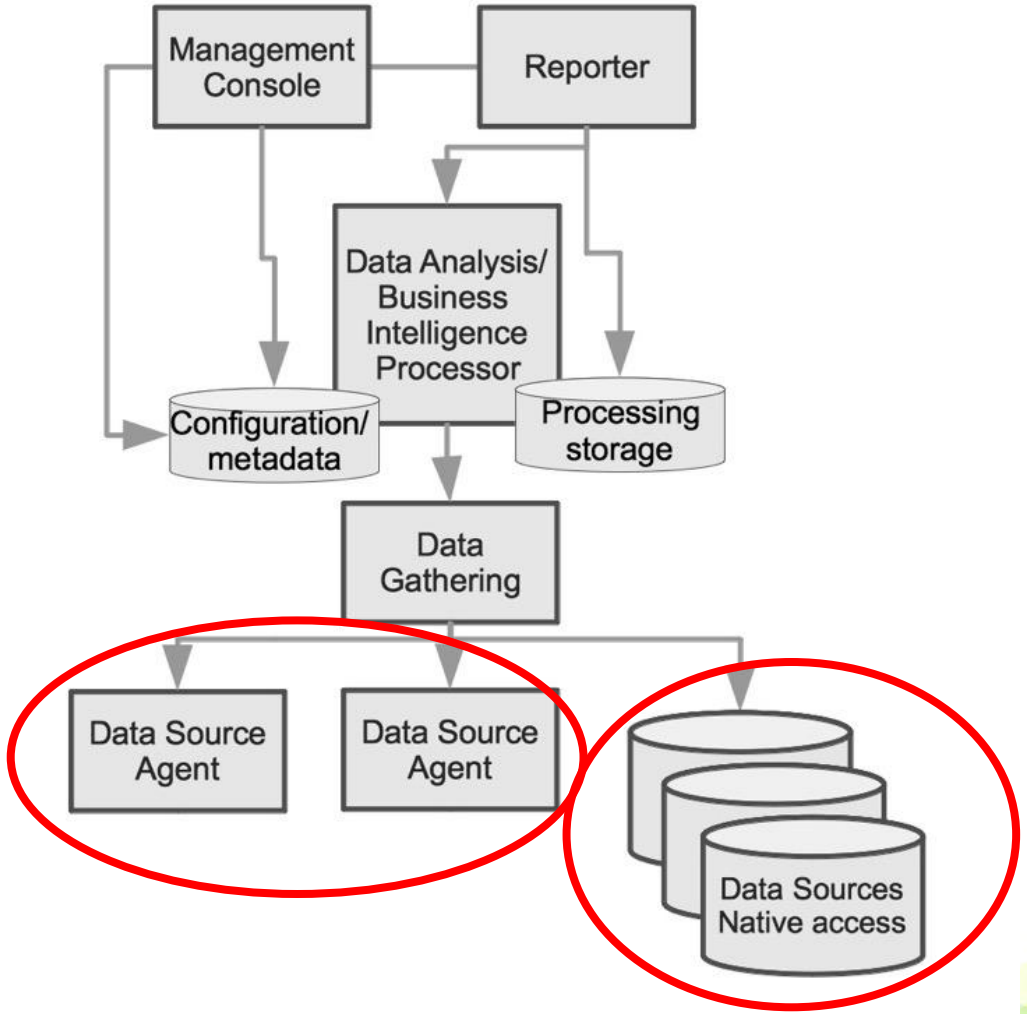
Analytics Data Harvesting Attack Surface



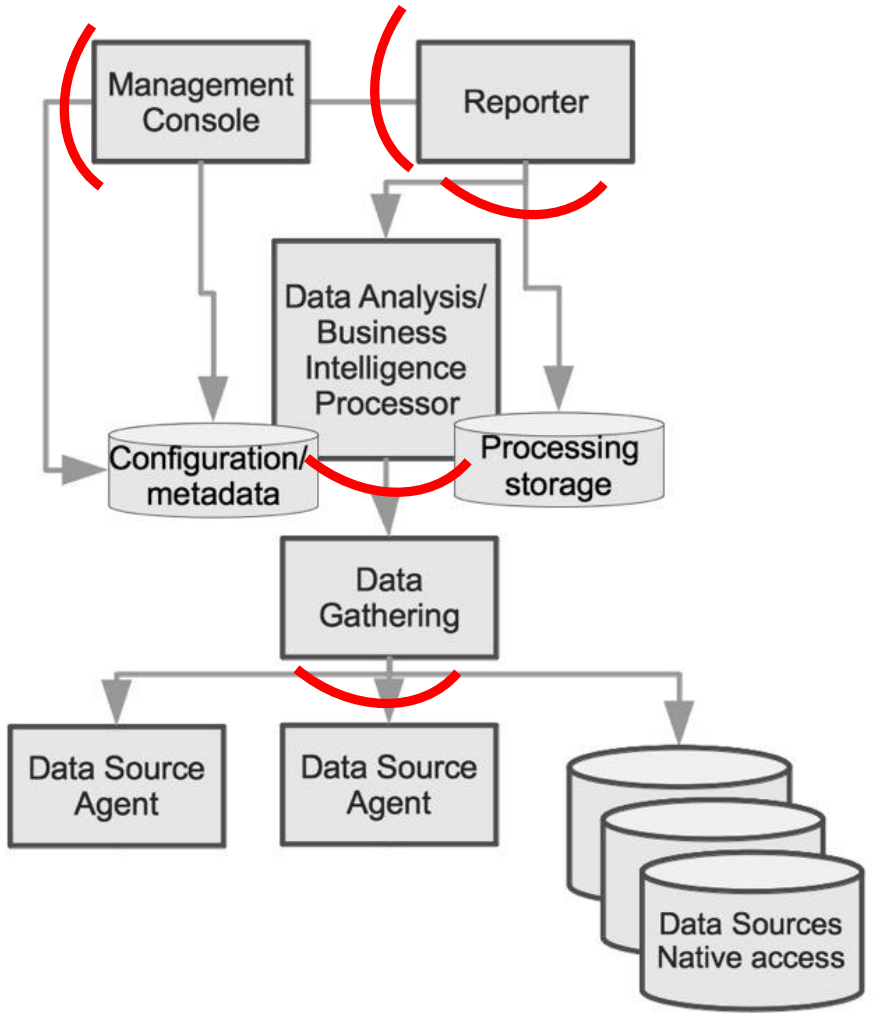
System Components



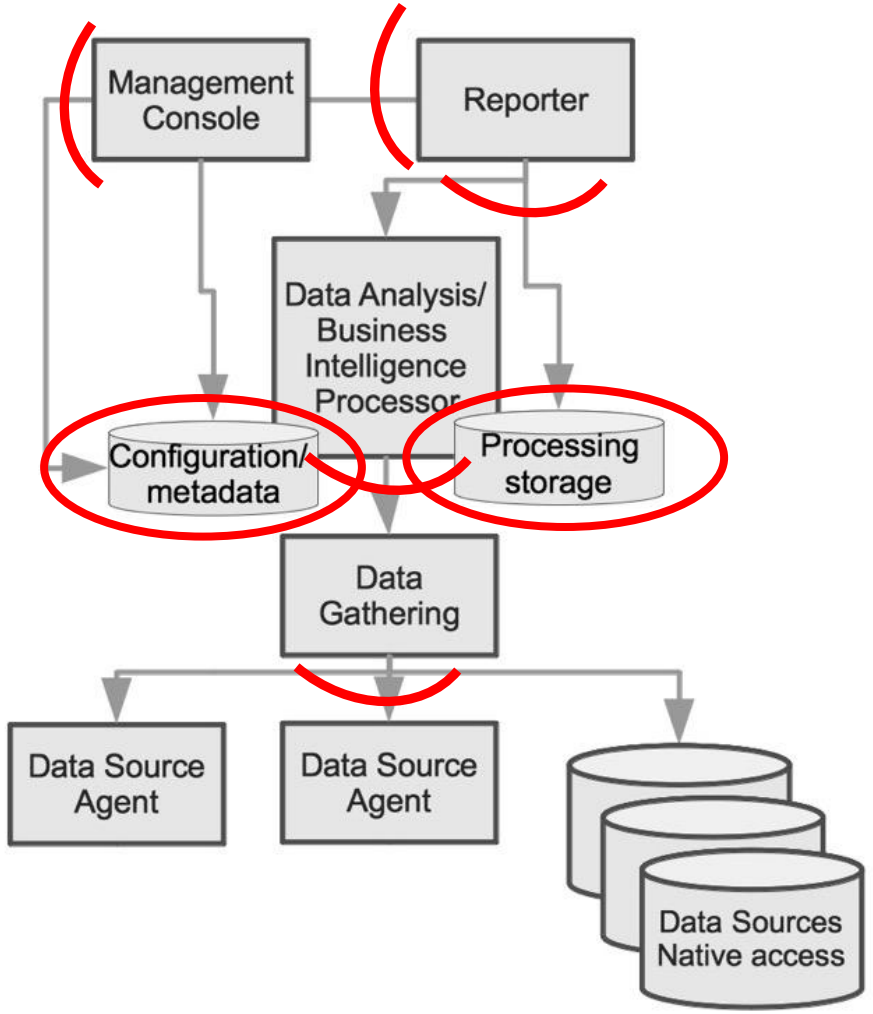
Holistic Analysis Will Protect Data Sources



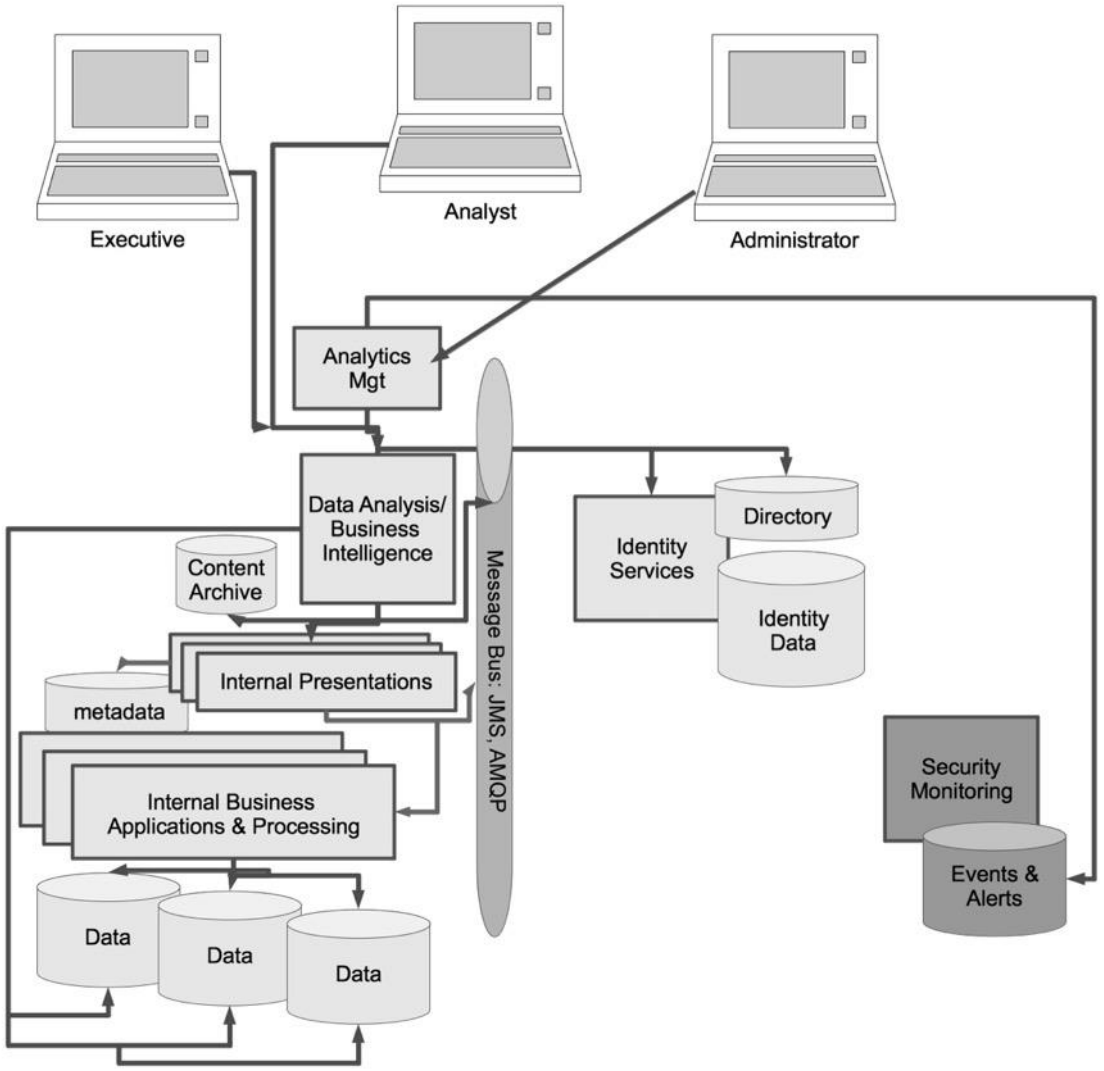
System Components Attack Surfaces



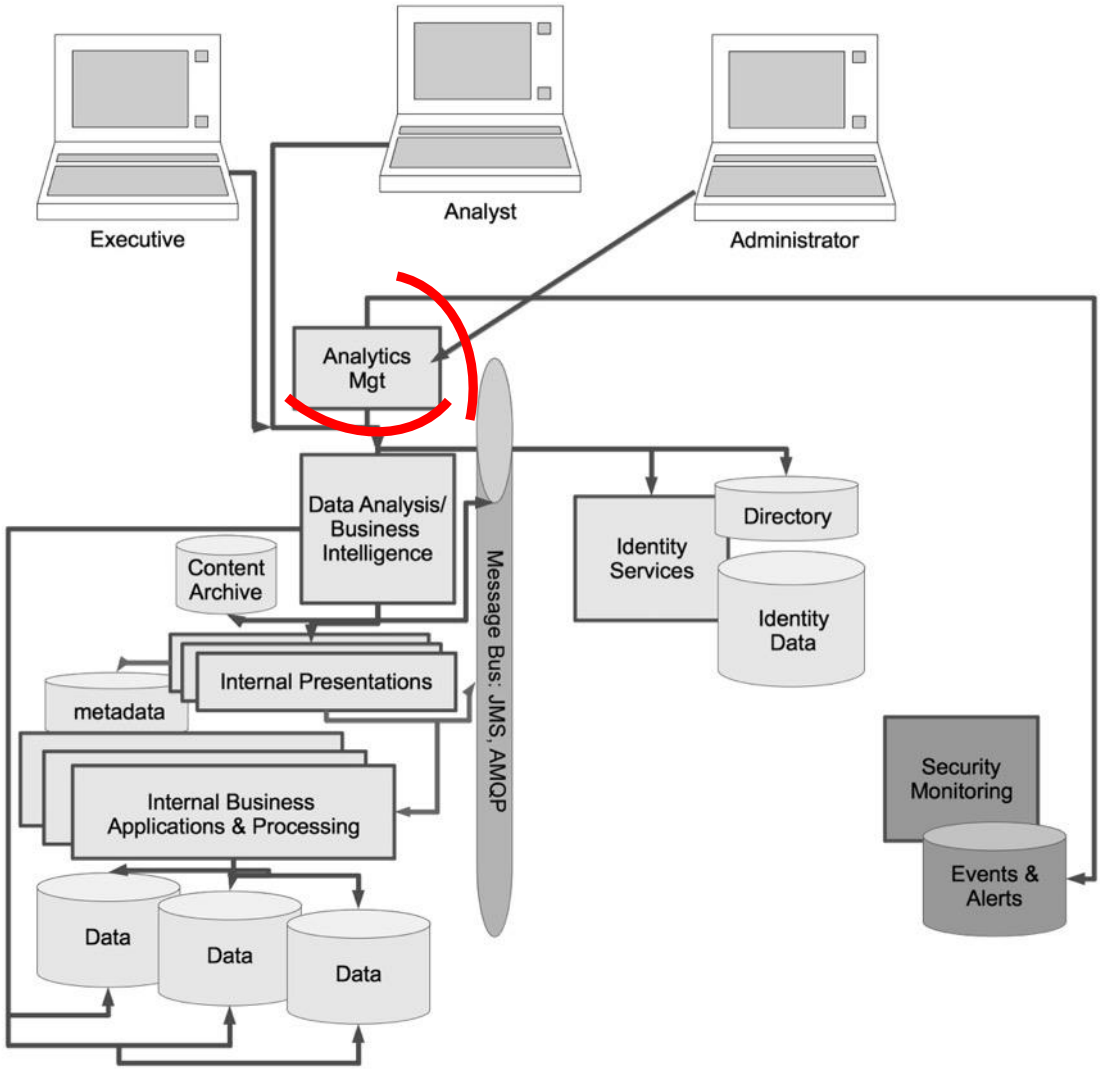
System Components Attacks At Assets



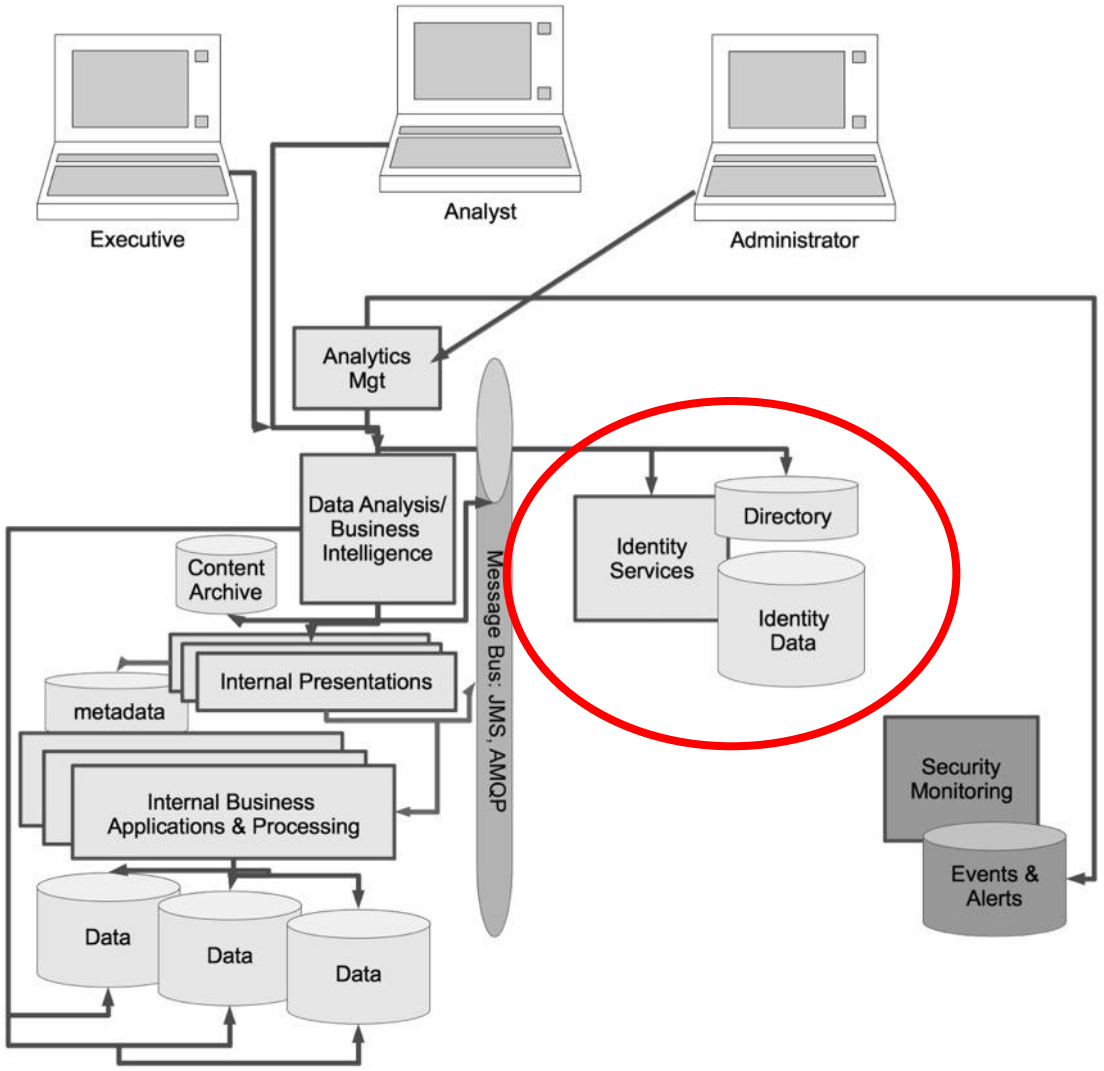
User Roles & Flows

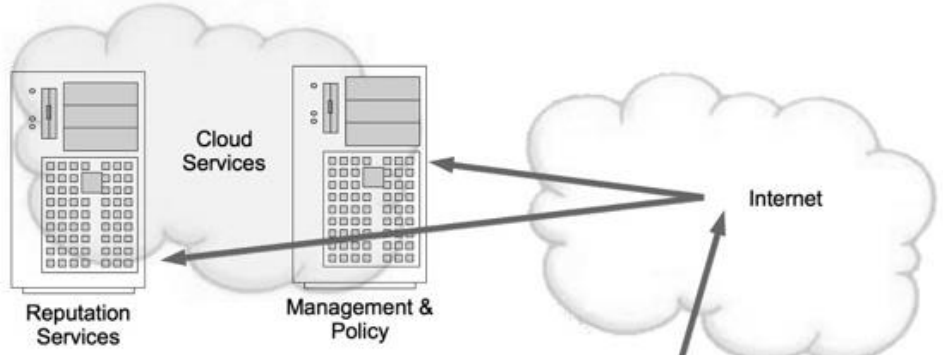


User Roles & Flows Attack Surfaces

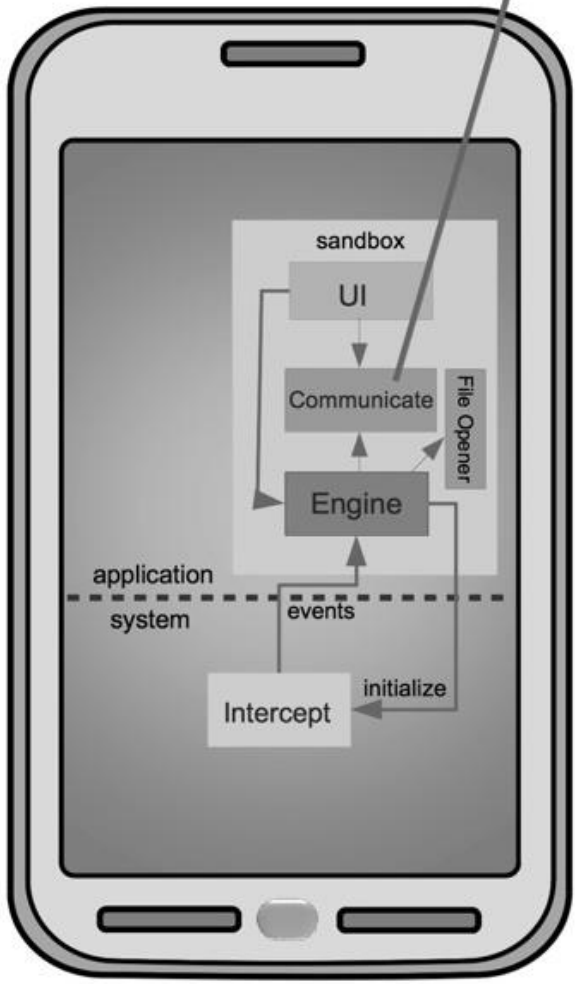


Identity Services: Who's In Danger?



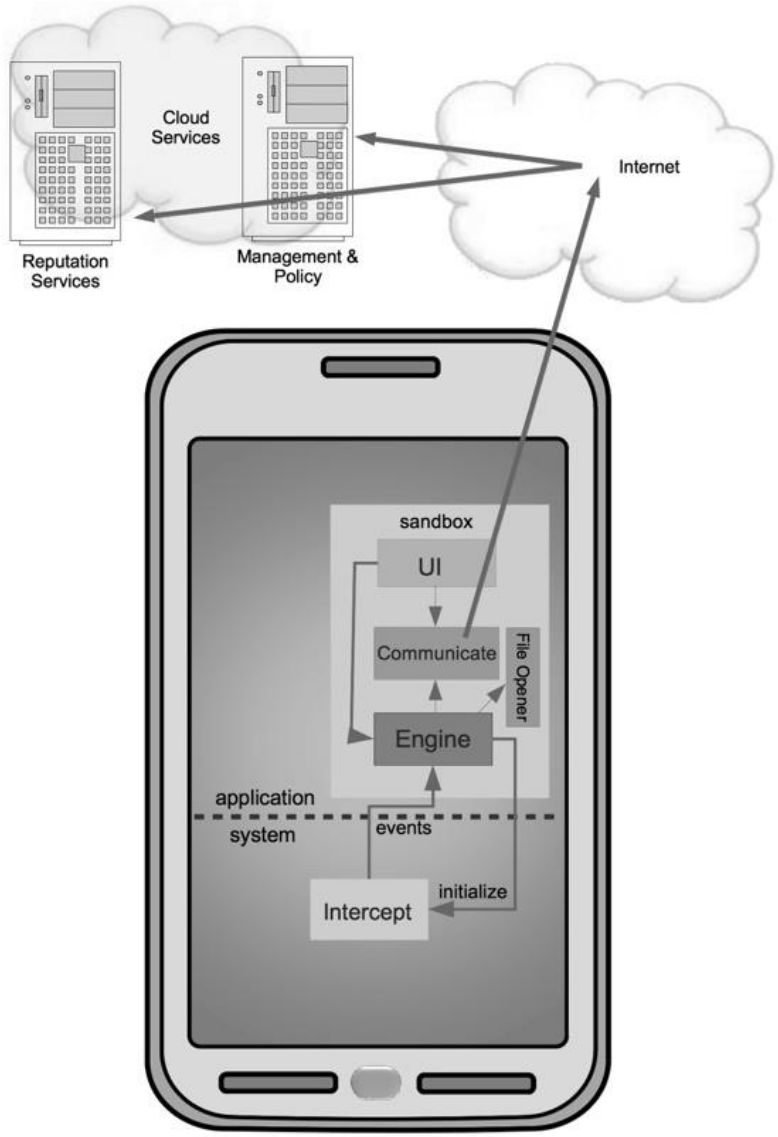


- Must reside in OS application sandbox
- Intercept in system to grab privileged events of interest for examination
- Intercept at higher privilege
- Intercept initialized early boot
- Intercept proxies events to engine
- Engine contains decision logic
- UI starts engine & communicate
- Files parsed/normalized by file opener
- External communications proxied through communicate
- Notifications from cloud through OS push notifications
- All message exchanges are initiated from device
    - Response to notification
- Device certificate/private key issued at enrollment
- Messages/updates signed by cloud services

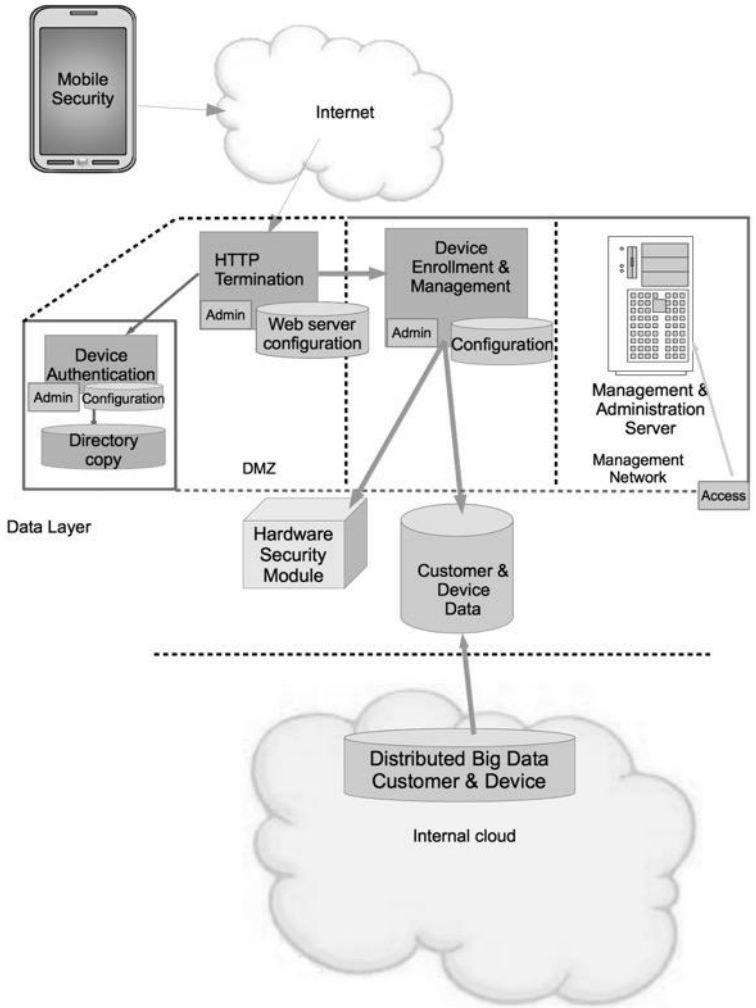


Mobile Security System

Mobile Security App
Break Into Teams To Analyze



Consider Server-side Components



After the threat model, then what?

Prioritization is one of the hardest problems

Simple Threat Model Steps (Won't likely be linear)

Enumerate CAV

Define and score impacts

Enumerate existing mitigations

Develop requirements to bring system to desired posture

Prioritize

Share and review with entire team + product management

How Do We Prioritize?

- Risk rate each CAV+impact that is not sufficiently protected
- Decide where the risk “line” is (low, medium, only high)
- Engage stakeholders in constructive dialog
- Write exceptions for items that should get implemented but must wait

Getting requirements implemented is an entirely different story…

(feel free to ask)

Effective Requirements

Getting requireme rent story…

(feel free to ask)

Composing requirements that can and will be built is a distinct discipline

Hints For Clearer Requirements

- Separate the functional need from any solution
  - Tell designers what function is required, not how to build that function
- Give implementers room for creativity and innovation
  - Don’t lock in obsolescence
- The amount of detail required is inverse to the amount of skill and knowledge of the implementers
  - Skilled teams need only understand what functionality is required
  - Unskilled implementers will need lots of detail

Closing thoughts

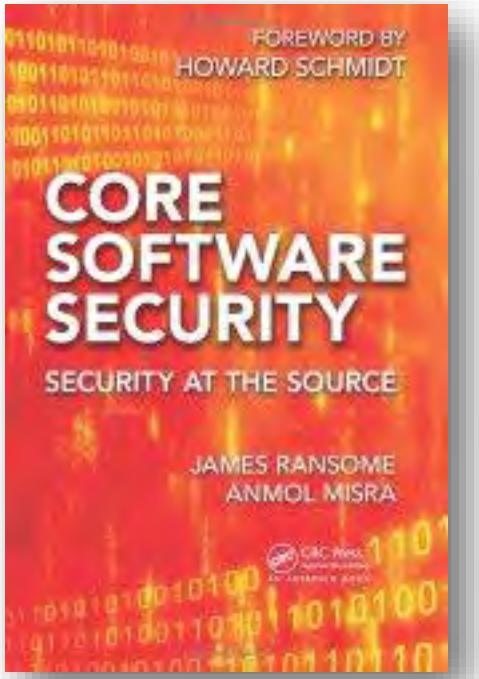
What kind of docs must you have?

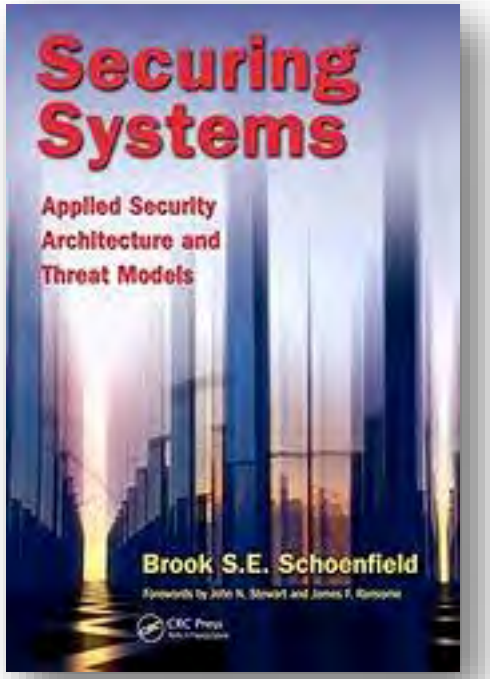
- The most important document is the requirements output from threat modeling
  - A threat model may be inferred from a thorough security requirements document
  - You must produce a requirements document
- If visual, one or more visual depictions of the architecture
  - Different domains often require different views onto the same system
  - Views might include:
    - attack surfaces
    - Assets
    - Mitigations and controls

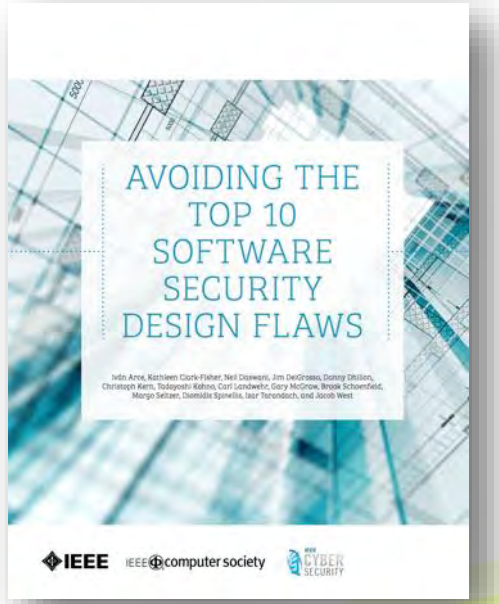
Next Steps

- Find an experienced mentor and 2 peer reviewers
  - Independence helps
- Just do it!
  - Threat models need to be revisited when architecture, threat landscape, and security features change
- Experienced threat modelers, please make yourself available!
- Start with your own projects
  - Gather the team and wrestle with the threat model
  - To gain experience, you can help with other threat models
    - Modeling diverse architecture deepens abilities

Shameless Self-promotion

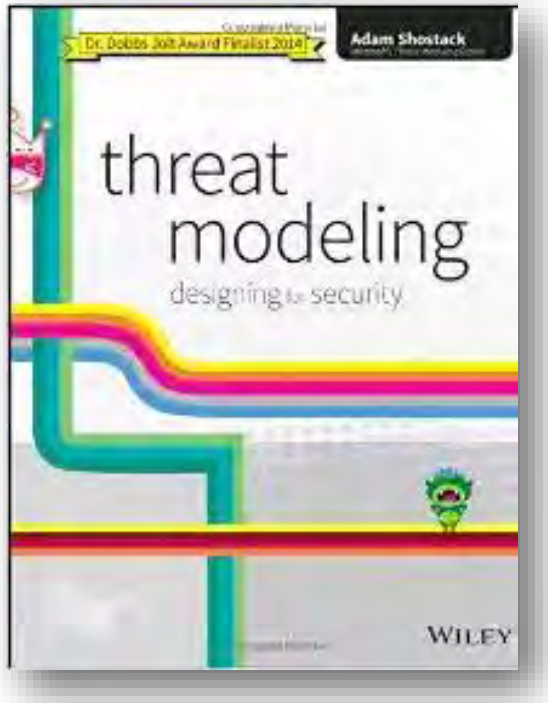


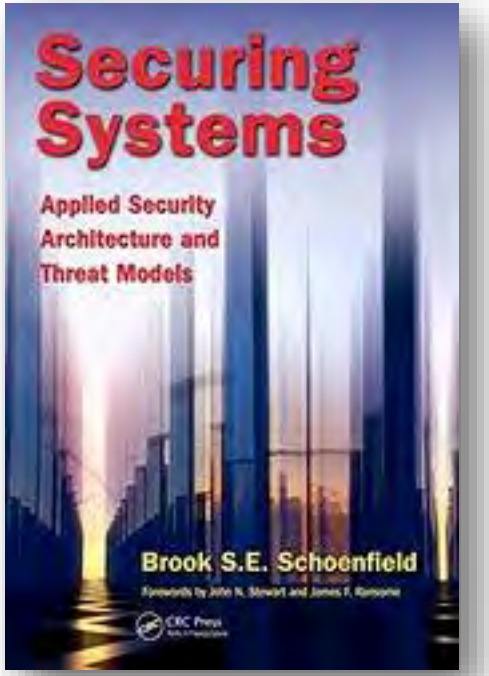


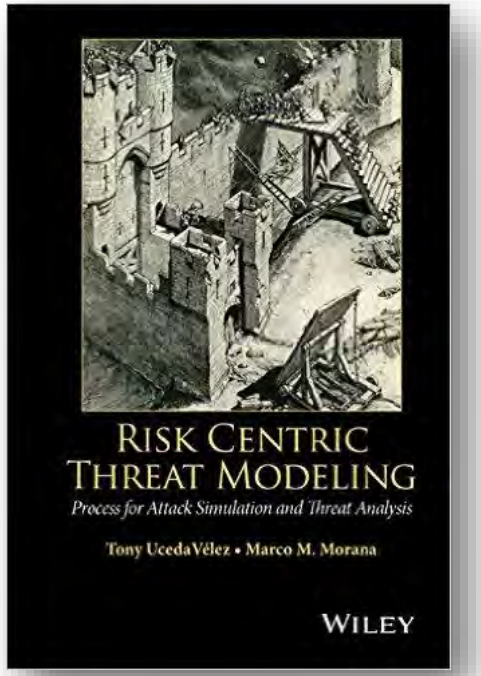


https://www.facebook.com/securingsystems

A Threat Modeling Library







https://www.facebook.com/securingsystems

Some Resources

https://www.owasp.org/index.php/Application_Threat_Modeling
https://www.owasp.org/index.php/Threat_Risk_Modeling
https://www.owasp.org/images/a/aa/AppSecEU2012_PASTA.pdf
http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=427321
http://www.intel.com/Assets/en_US/PDF/whitepaper/wp_IT_Security_RiskAssessment.pdf
https://www.facebook.com/securingsystems
http://www.amazon.com/Securing-Systems-Applied-Security-Architecture/dp/1482233975
https://www.facebook.com/softwaresec
www.amazon.com/Core-Software-Security-Source

Where To Find Me

Brook.e.schoenfield@intel.com

http://www.brookschoenfield.com

brook@brookschoenfield.com

@BrkSchoenfield

1. I apologize in advance. I only Linkedin with people with whom I've had meaningful interaction. Thanks.

Brook's Social Networking

https://www.linkedin.com/in/brookschoenfield [1]

http://www.amazon.com/Brook-S.-E.-Schoenfield/e/B00XQFZLSW

1. I apologize in advance. I only Linkedin with people with whom I've had meaningful interaction. Thanks.

Q & A

www.brookschoenfield.com

A Threat Agent Matrix

Threat Agent	Goals	Technical Ability	Risk Tolerance	Work Factor	Activity Level
Cybercrime	Monetary	Low (known proven)	Low to medium	low	Very high, continual
Industrial Espionage	Information	Medium to medium-high	low	medium	Low. For enterprises, medium
Nation-states	Information Disruption	Very high	Very low	Very high	Medium but constant
Law Enforcement/Gov Compliance	Compliance Information	Medium	None – they are the law	medium	intermittent
Insider	monetary	Varies	Low	none	Occasional
Insider	Revenge	Varies	Very high	none	Occasional
Usage abuse	Unauthorized use	Low	Low	Low	constant
Hacktivists	Media attention for cause	Low to medium	Used to be high, now much lower	medium	intermittent
Hackers	Status	Often very low	Low	low	low
Security Researcher	Career enhancement	High	None	high	medium