#DOXNYC
DevOps Exchange New York City

DevOps Exchange 2021: Security Champions

Presented by Rick Payne MSc, CISSP, Security+ , BS-IST, AS-CET, AWS CSA, RHCE, RHCSA

About me

CAREER	CONF/PRES	EDUCATION
Principal Security Engineer	+1 DOX ☺	
Staff Security Engineer	>35	MSc
Sr. Security Engineer		
Chief Security Officer		AWS-CSA
Security Architect		CISSP
Security Analyst I/II	^^^	BS, AA, Security+, RHCE, RHCSA
System Integration Tech I/II		
Manufacturing Test Tech I/II		AS-CET
Production Test Intern		

Presentation Goals

Security Responsibility Awareness – Who's role is it anyway?

Maturity Model – Where am I? Where do I need to go?

Actionable – Start securing your systems today!

2021 GitLab DevSecOps Survey

Sec and dev are friendlier, but there is still confusion over who "owns" security, and the finger-pointing game is strong[1]

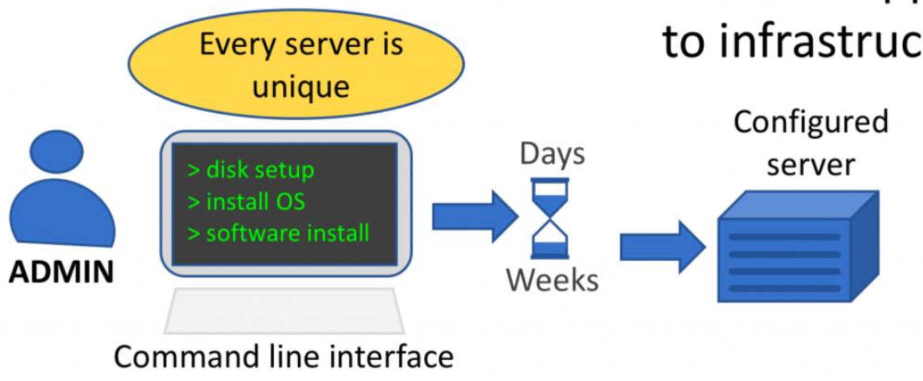
A full 39% of developers feel fully responsible for security in their organizations (up from 28% last year), while 32% said they shared the burden with other teams[1]

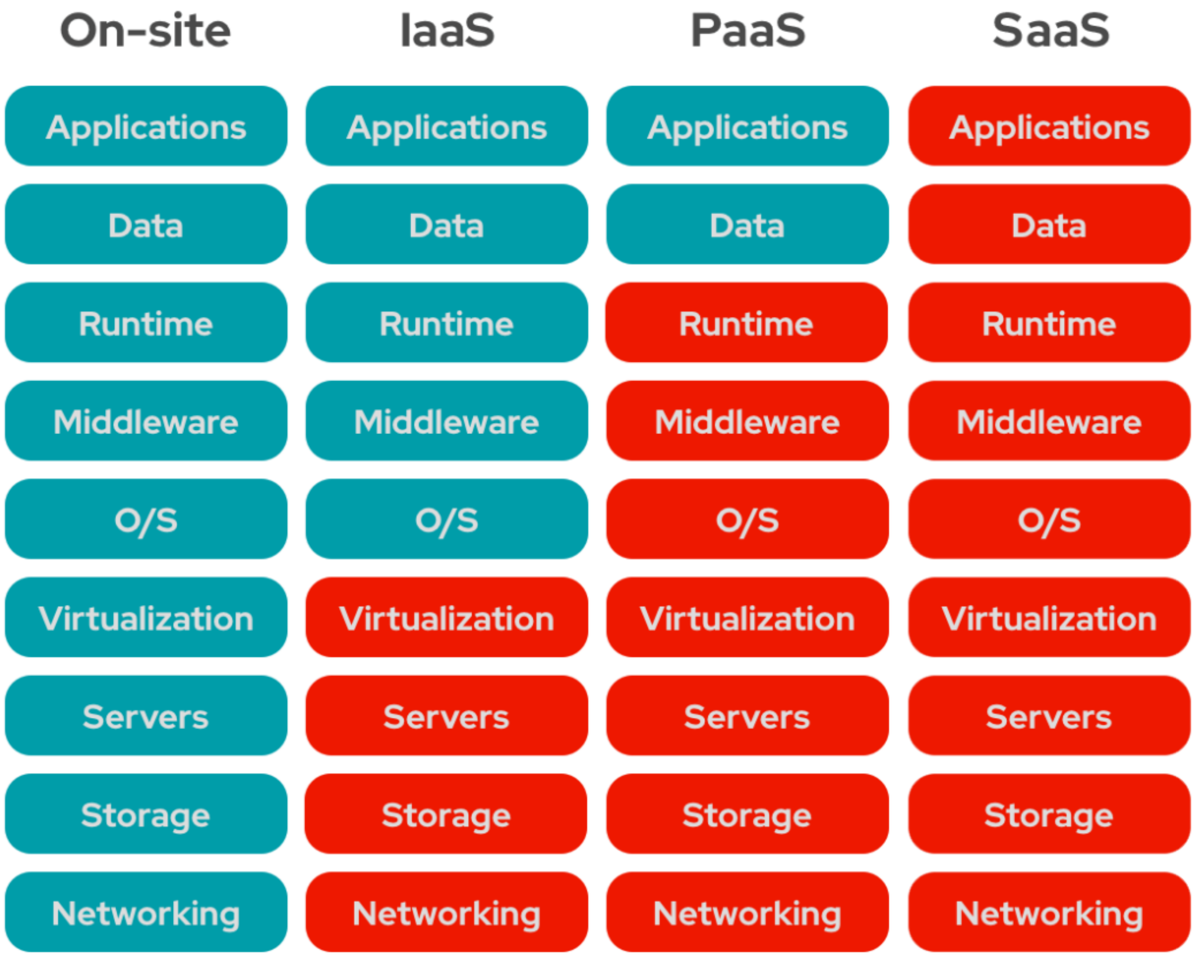
Dev-centric questions with app-focused answers. What about the Infra & Platform layers?

Shifting Left: R&Rs Follow the Code!

Land before time

Traditional approach to infrastructure





On-site	IaaS	PaaS	SaaS
Applications	Applications	Applications	Applications
Data	Data	Data	Data
Runtime	Runtime	Runtime	Runtime
Middleware	Middleware	Middleware	Middleware
O/S	O/S	O/S	O/S
Virtualization	Virtualization	Virtualization	Virtualization
Servers	Servers	Servers	Servers
Storage	Storage	Storage	Storage
Networking	Networking	Networking	Networking

🟦 You manage

🟥 Service provider manages

Shifting Left: R&Rs Then -> Now

Requirements / Scan Result Punt -> Security Engineering

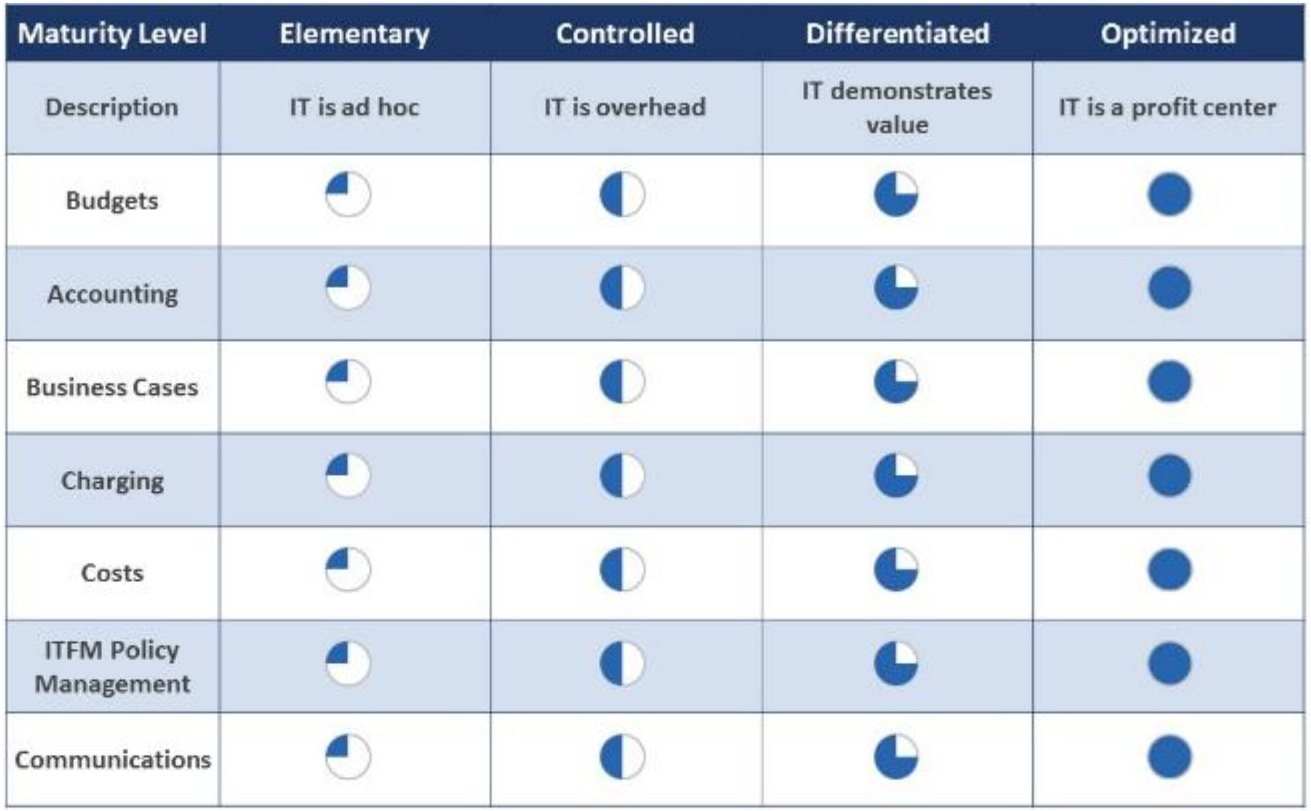
Fundamental Human Capacity Problem

1.5 security experts per 100 software engineers[2]

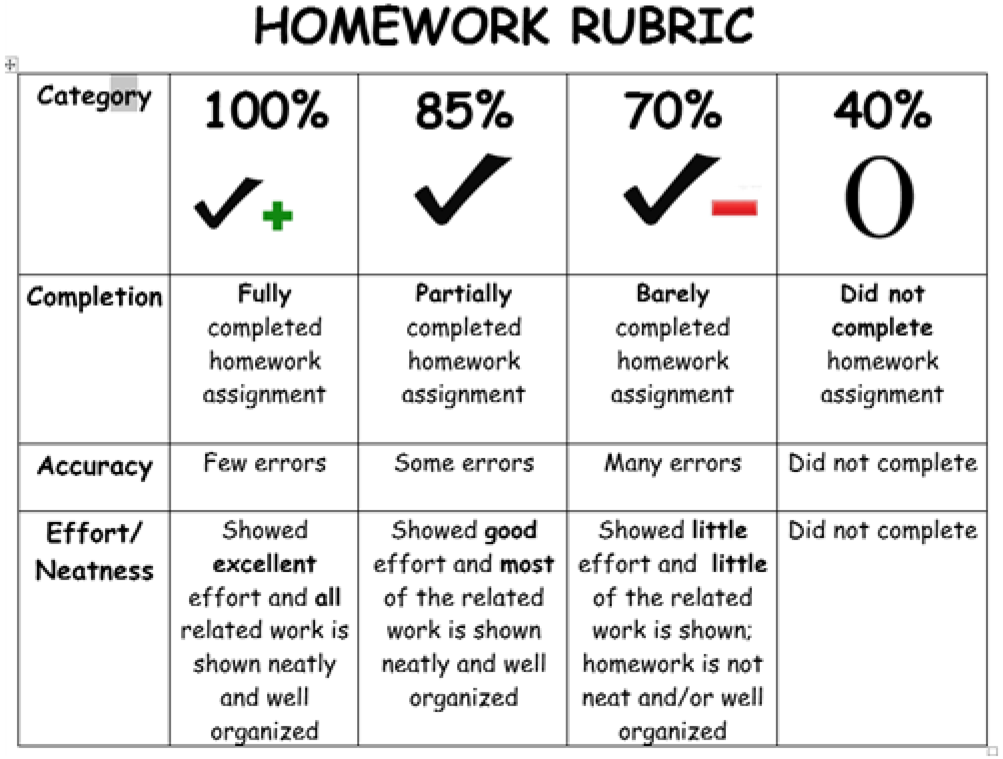
Day 1 Cybersecurity Debt

What are Maturity Models?

- Essentially, how mature is an organization at a specific objective.



Maturity Level	Elementary	Controlled	Differentiated	Optimized
Description	IT is ad hoc	IT is overhead	IT demonstrates value	IT is a profit center
Budgets	◔	◐	◕	●
Accounting	◔	◐	◕	●
Business Cases	◔	◐	◕	●
Charging	◔	◐	◕	●
Costs	◔	◐	◕	●
ITFM Policy Management	◔	◐	◕	●
Communications	◔	◐	◕	●

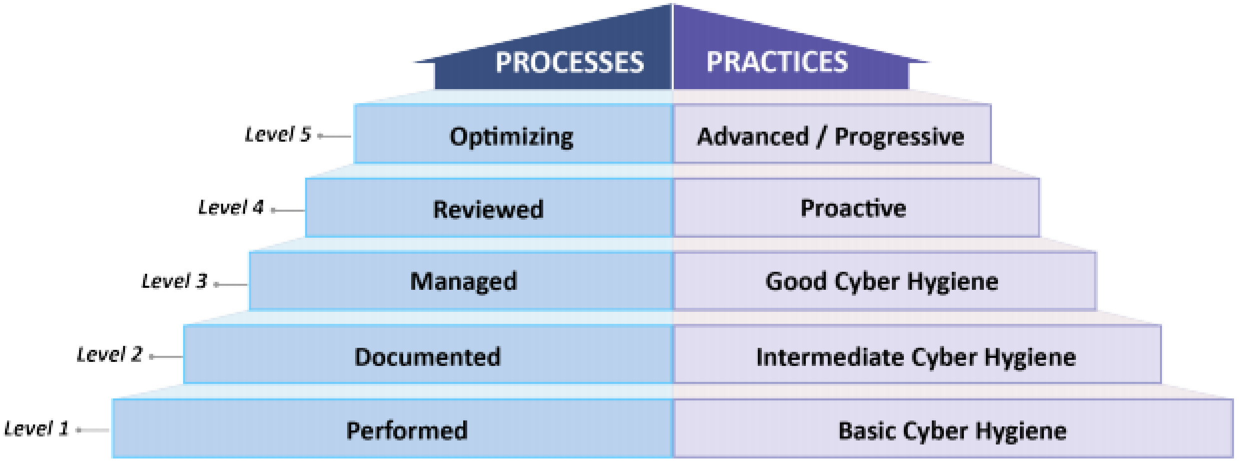


HOMEWORK RUBRIC

Category	100% ✓+	85% ✓	70% ✓−	40% O
Completion	**Fully** completed homework assignment	**Partially** completed homework assignment	**Barely** completed homework assignment	**Did not complete** homework assignment
Accuracy	Few errors	Some errors	Many errors	Did not complete
Effort/ Neatness	Showed **excellent** effort and **all** related work is shown neatly and well organized	Showed **good** effort and **most** of the related work is shown neatly and well organized	Showed **little** effort and **little** of the related work is shown; homework is not neat and/or well organized	Did not complete

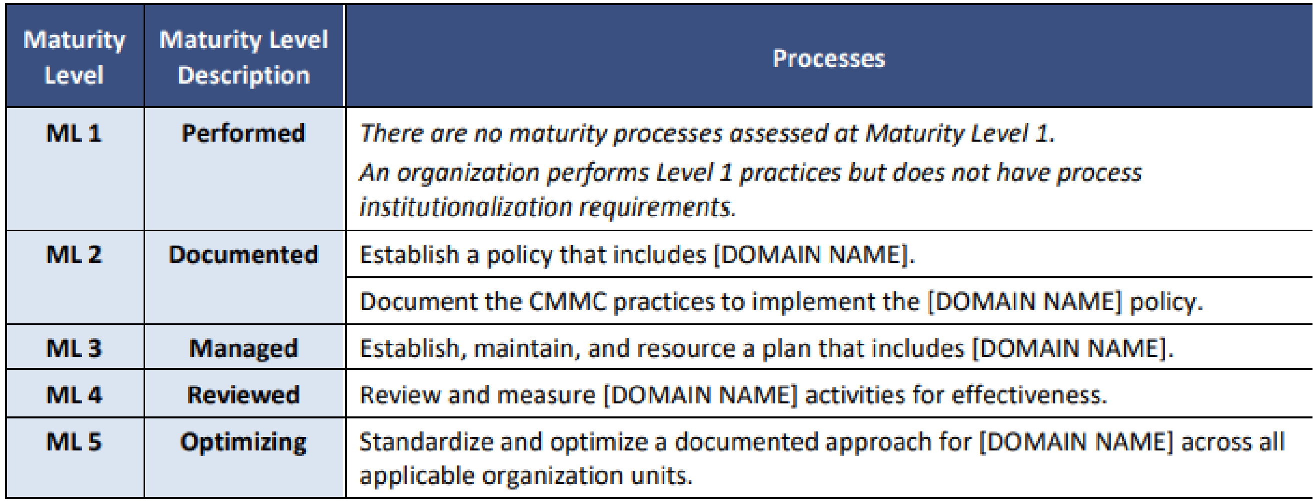
What are Maturity Models?

- DoD’s Cybersecurity Maturity Model example



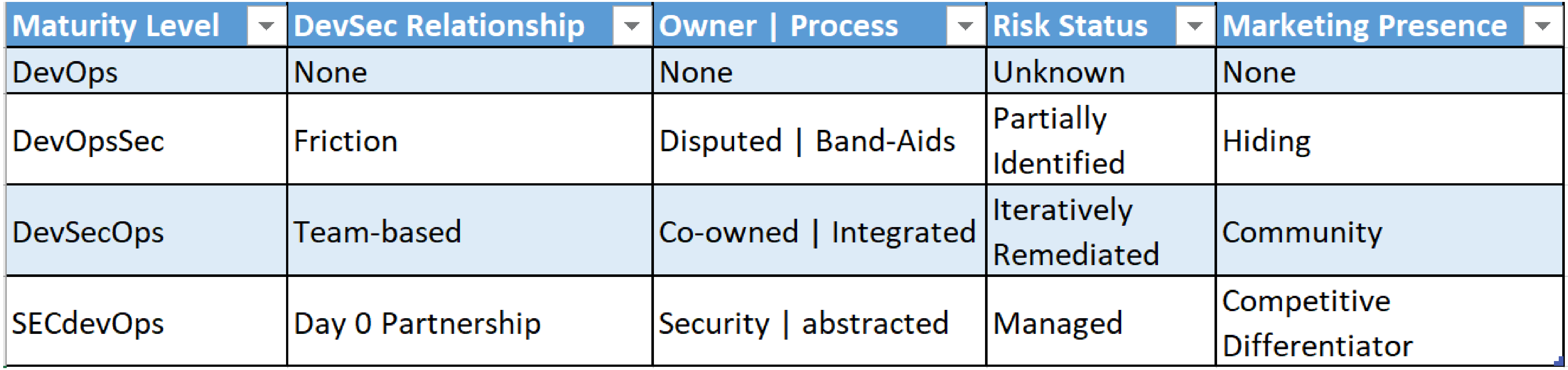
	PROCESSES	PRACTICES
Level 5	Optimizing	Advanced / Progressive
Level 4	Reviewed	Proactive
Level 3	Managed	Good Cyber Hygiene
Level 2	Documented	Intermediate Cyber Hygiene
Level 1	Performed	Basic Cyber Hygiene

Figure 2. CMMC Levels and Descriptions



Maturity Level	Maturity Level Description	Processes
ML 1	**Performed**	*There are no maturity processes assessed at Maturity Level 1.* *An organization performs Level 1 practices but does not have process institutionalization requirements.*
ML 2	**Documented**	Establish a policy that includes [DOMAIN NAME].
		Document the CMMC practices to implement the [DOMAIN NAME] policy.
ML 3	**Managed**	Establish, maintain, and resource a plan that includes [DOMAIN NAME].
ML 4	**Reviewed**	Review and measure [DOMAIN NAME] activities for effectiveness.
ML 5	**Optimizing**	Standardize and optimize a documented approach for [DOMAIN NAME] across all applicable organization units.

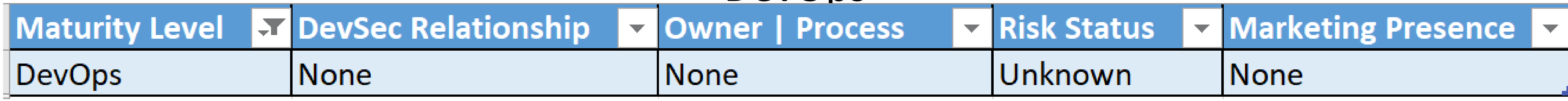
Maturity Models – DevOps -> SECdevOps



Maturity Level	DevSec Relationship	Owner | Process	Risk Status	Marketing Presence
DevOps	None	None	Unknown	None
DevOpsSec	Friction	Disputed | Band-Aids	Partially Identified	Hiding
DevSecOps	Team-based	Co-owned | Integrated	Iteratively Remediated	Community
SECdevOps	Day 0 Partnership	Security | abstracted	Managed	Competitive Differentiator

Maturity Models – DevOps -> SECdevOps

- DevOps

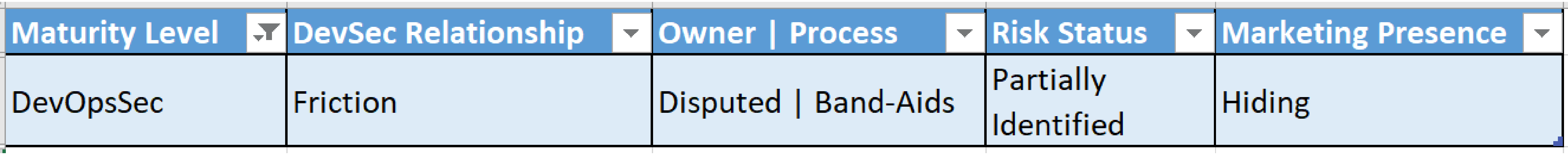


Maturity Level	DevSec Relationship	Owner | Process	Risk Status	Marketing Presence
DevOps	None	None	Unknown	None

- DevOps without security is essentially a product pen test sandbox!
- Most default deployments can consist of 100s or >1000s configurations or vulnerabilities
- Business perspective -> You can’t acquire or maintain any security certifications

Maturity Models – DevOps -> SECdevOps

- DevOpsSec <- Shift left

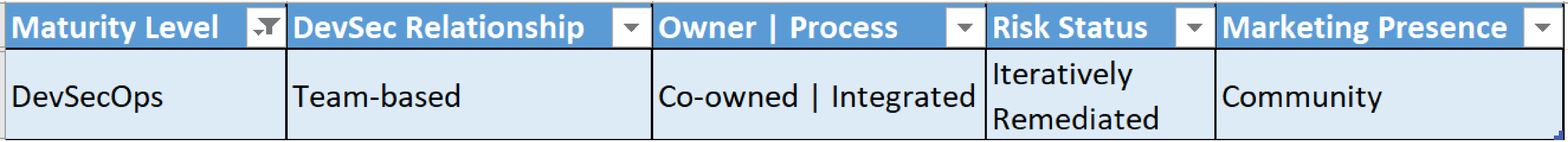


Maturity Level	DevSec Relationship	Owner | Process	Risk Status	Marketing Presence
DevOpsSec	Friction	Disputed | Band-Aids	Partially Identified	Hiding

- Security is begrudgingly forced in
- Still political friction
- Security is chasing production / already released workloads

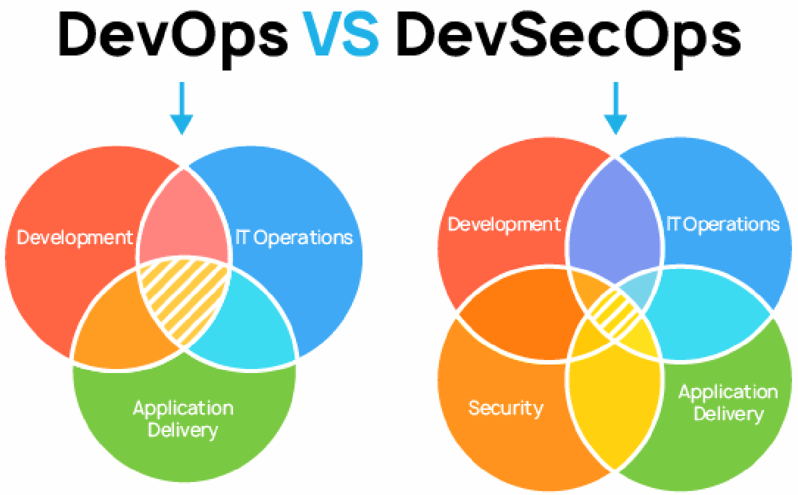
Maturity Models – DevOps -> SECdevOps

- DevSecOps <- Shift left



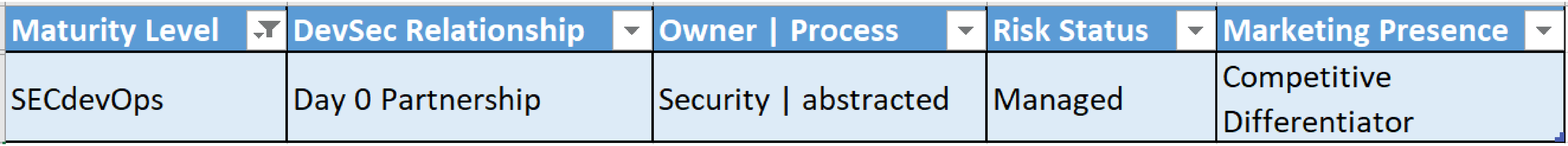
Maturity Level	DevSec Relationship	Owner | Process	Risk Status	Marketing Presence
DevSecOps	Team-based	Co-owned | Integrated	Iteratively Remediated	Community

- Culture shift:
  - Roles: We’re a team!
  - Responsibilities: Every piece of the puzzle has a security layer with support



Maturity Models – DevOps -> SECdevOps

- SECdevOps

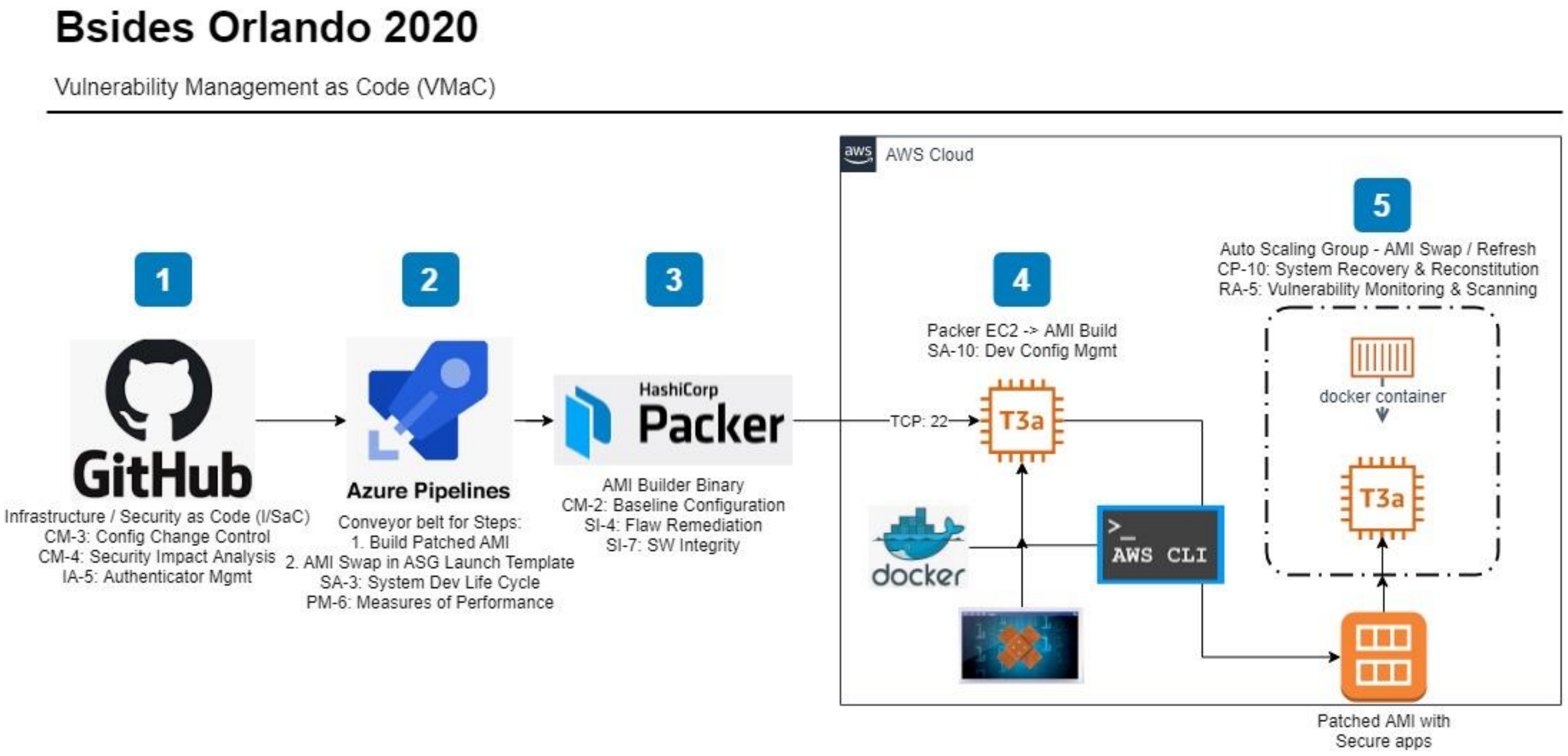


Maturity Level	DevSec Relationship	Owner | Process	Risk Status	Marketing Presence
SECdevOps	Day 0 Partnership	Security | abstracted	Managed	Competitive Differentiator

- Security is at the forefront of everything
  - Project kickoffs
  - Design reviews
- Devs can now focus on secure code and update support as Infra & Platform are invisible. Security becomes FREE!
- Security features are equally or more important than dev

Demo - Lightweight, Conceptual

- Vulnerability Management as Code (VMaC) solution overview



Bsides Orlando 2020

Vulnerability Management as Code (VMaC)

1
GitHub
Infrastructure / Security as Code (I/SaC)
CM-3: Config Change Control
CM-4: Security Impact Analysis
IA-5: Authenticator Mgmt

2
Azure Pipelines
Conveyor belt for Steps:
1. Build Patched AMI
2. AMI Swap in ASG Launch Template
SA-3: System Dev Life Cycle
PM-6: Measures of Performance

3
HashiCorp Packer
AMI Builder Binary
CM-2: Baseline Configuration
SI-4: Flaw Remediation
SI-7: SW Integrity

AWS Cloud

4
Packer EC2 -> AMI Build
SA-10: Dev Config Mgmt
TCP: 22
T3a
docker
AWS CLI

5
Auto Scaling Group - AMI Swap / Refresh
CP-10: System Recovery & Reconstitution
RA-5: Vulnerability Monitoring & Scanning
docker container
T3a

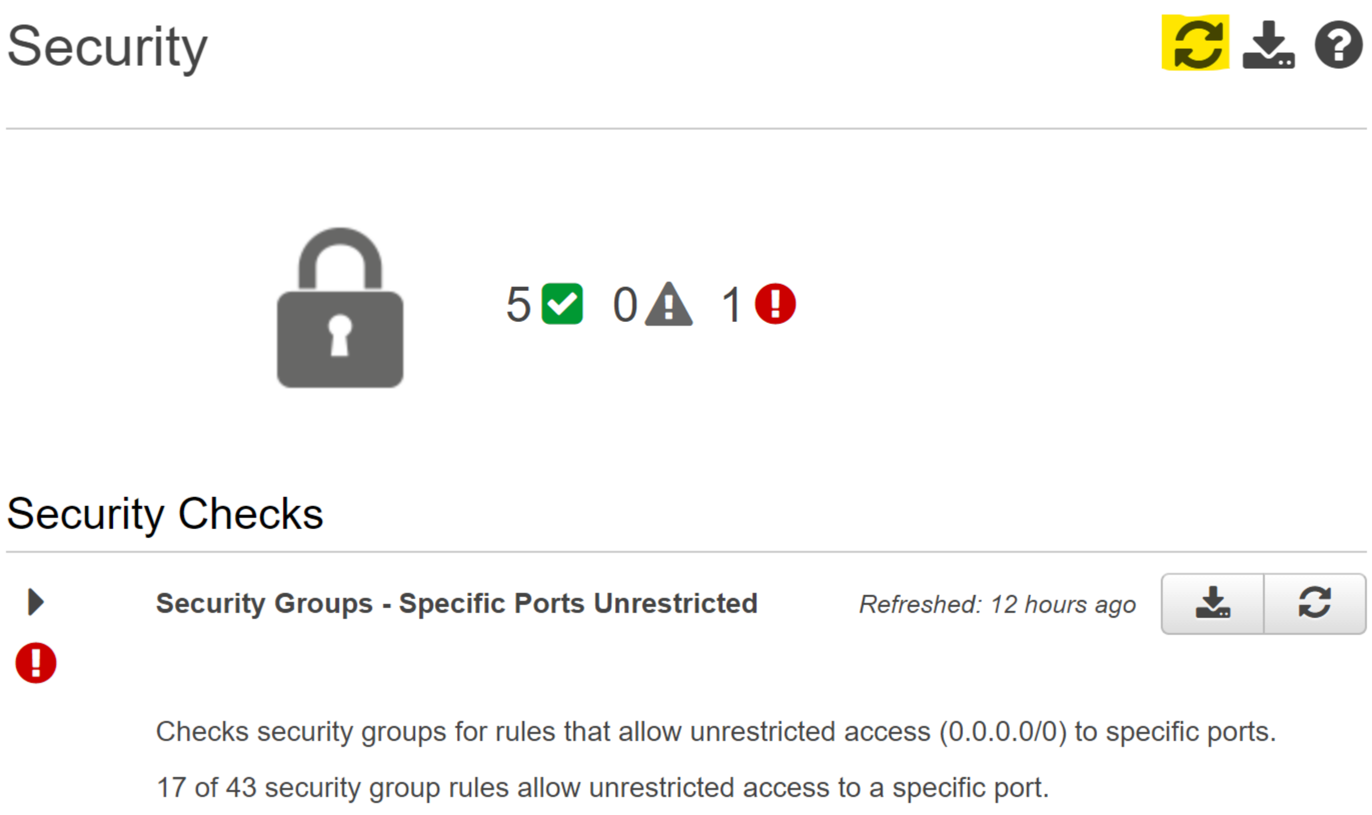
Patched AMI with Secure apps

Demo – IDE Scanning – The Source

- Visual Studio Code – Snyk Vulnerability Scanner https://marketplace.visualstudio.com/items?itemName=snyk-security.snyk-vulnerability-scanner
- Visual Studio Code – Snyk Vuln Cost https://marketplace.visualstudio.com/items?itemName=snyk-security.vscode-vuln-cost

Demo – AWS Trusted Advisor

- Vulnerability Management as Code (VMaC) solution overview



Security

5 ✓ 0 ⚠ 1 ❗

Security Checks

Security Groups - Specific Ports Unrestricted *Refreshed: 12 hours ago*

Checks security groups for rules that allow unrestricted access (0.0.0.0/0) to specific ports.

17 of 43 security group rules allow unrestricted access to a specific port.

Demo – DoD Secure Configuration Guidance

- STIG Viewer - https://public.cyber.mil/stigs/stig-viewing-tools/
- STIG Library - https://public.cyber.mil/stigs/compilations/

Kubernetes: 93 (15 Sev 1 e.g. The Kubernetes Kubelet must have the read-only port flag disabled.)

Server 2019: 304 (33 Sev 1 e.g. Windows Server 2019 must be configured to prevent the storage of the LAN Manager hash of passwords.)

Red Hat 8: 343 (18 Sev 1 e.g. The root account must be the only account having unrestricted access to the RHEL 8 system.)

Ubuntu 20: 185 (11 Sev 1 e.g. The Ubuntu operating system must not have the telnet package installed.)

MS SQL 2016: 144 (12 Sev 1 e.g. When using command-line tools such as SQLCMD in a mixed-mode authentication environment, users must use a logon method that does not expose the password.)

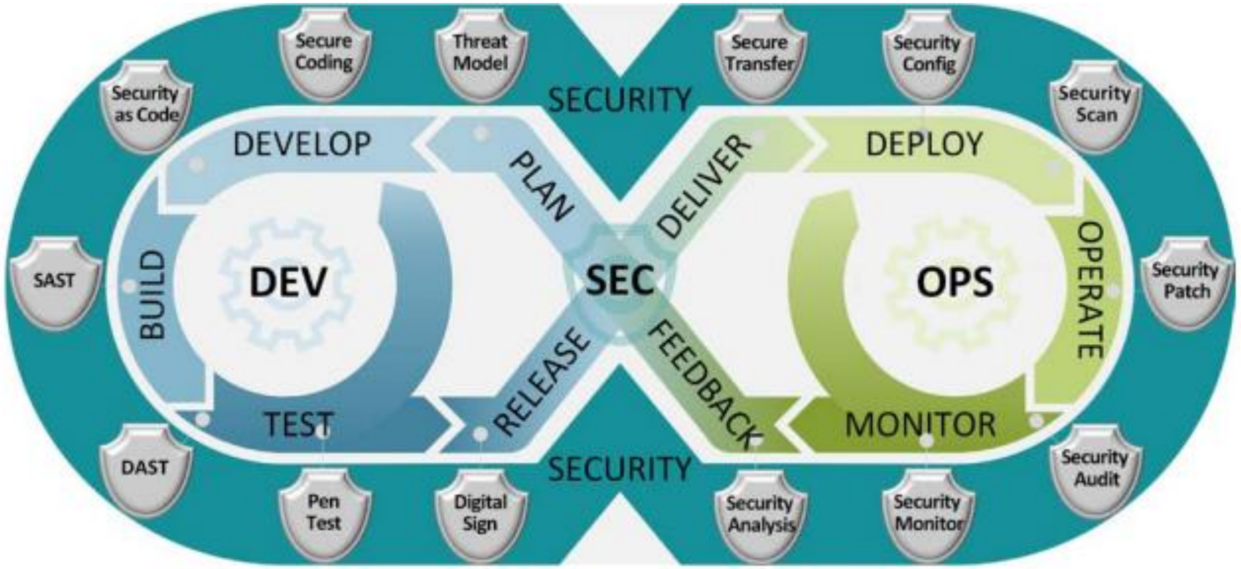
Apache 2.4: 75 (5 Sev 1 e.g. The account used to run the Apache web server must not have a valid login shell and password defined.)

1144 total, 94 Sev 1

NIST 800-53r4 – Table H-1 to ISO 27001 - https://csrc.nist.gov/publications/detail/sp/800-53/rev-4/final

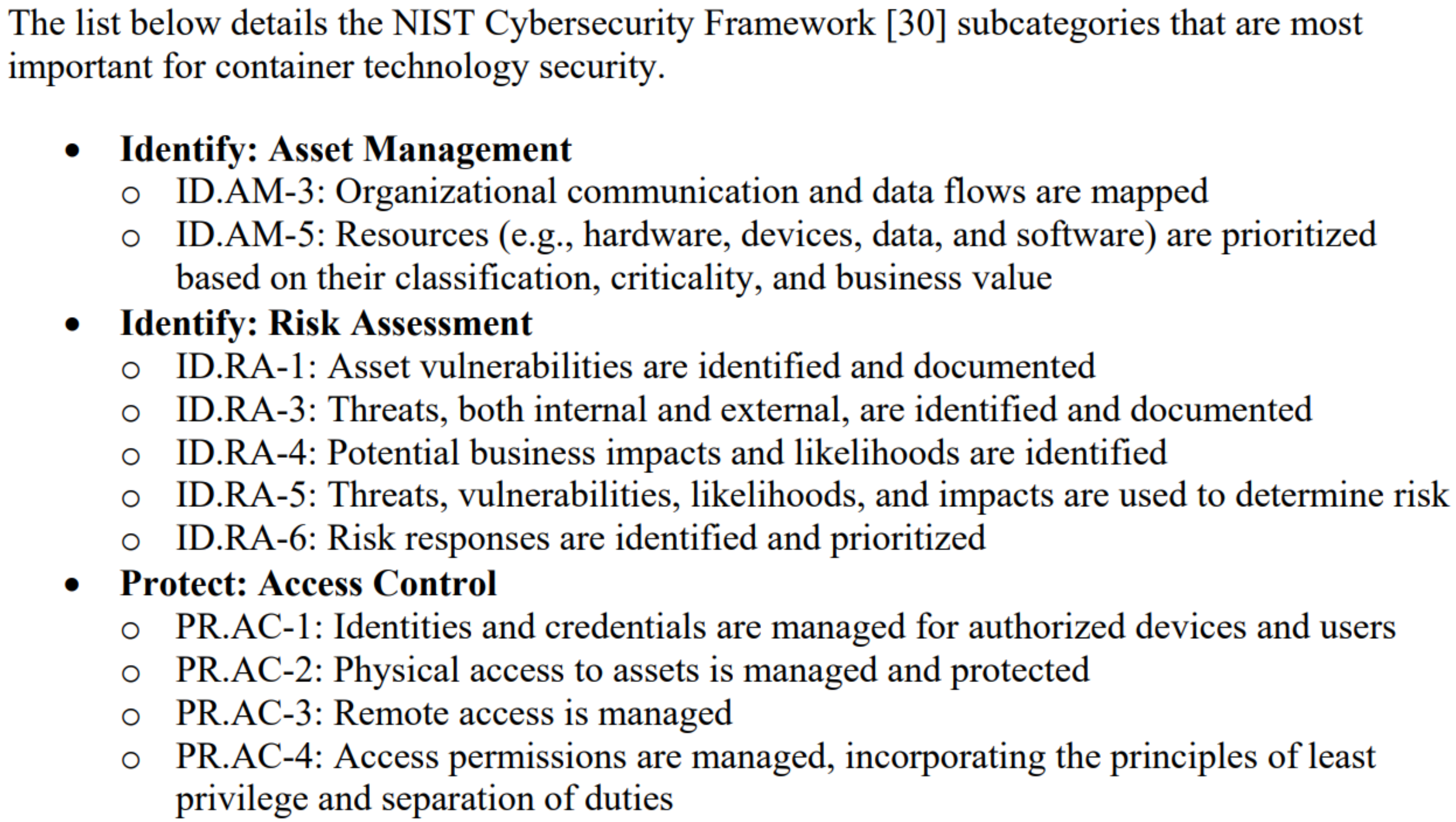
Demo – DoD Secure Configuration Guidance

- Free OS Configuration Scanning - https://public.cyber.mil/stigs/scap/
- Container Hardening Guide V1R1 - https://software.af.mil/wp-content/uploads/2021/03/2020_Oct_-Final-DevSecOps-Enterprise-Container-Hardening-Guide-1.1-Public-Release.pdf
- DevSecOps Reference Architecture - https://dodcio.defense.gov/Portals/0/Documents/DoD%20Enterprise%20DevSecOps%20Reference%20Design%20v1.0_Public%20Release.pdf?ver=2019-09-26-115824-583



Demo – NIST Container Security

- Application Container Security Guide - https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-190.pdf



The list below details the NIST Cybersecurity Framework [30] subcategories that are most important for container technology security.

- **Identify: Asset Management**
  - ID.AM-3: Organizational communication and data flows are mapped
  - ID.AM-5: Resources (e.g., hardware, devices, data, and software) are prioritized based on their classification, criticality, and business value
- **Identify: Risk Assessment**
  - ID.RA-1: Asset vulnerabilities are identified and documented
  - ID.RA-3: Threats, both internal and external, are identified and documented
  - ID.RA-4: Potential business impacts and likelihoods are identified
  - ID.RA-5: Threats, vulnerabilities, likelihoods, and impacts are used to determine risk
  - ID.RA-6: Risk responses are identified and prioritized
- **Protect: Access Control**
  - PR.AC-1: Identities and credentials are managed for authorized devices and users
  - PR.AC-2: Physical access to assets is managed and protected
  - PR.AC-3: Remote access is managed
  - PR.AC-4: Access permissions are managed, incorporating the principles of least privilege and separation of duties

Demo – K8s Secure Configuration Guidance

- DoD K8s STIG Export
- AWS EKS Security - https://docs.aws.amazon.com/eks/latest/userguide/security.html
- EKS Scanning with kube-bench - https://aws.amazon.com/blogs/containers/introducing-cis-amazon-eks-benchmark/
  - 42m00s - https://www.eksworkshop.com/intermediate/300_cis_eks_benchmark/

Last but not least, take care of yourself!

Q&A / References

- GitHub - https://github.com/rickpayne929/presentations
- Event Page - https://www.meetup.com/DOXNYC/events/278299834/
- DoD Cybersecurity Maturity Model Certification (CMMC) - https://www.acq.osd.mil/cmmc/docs/CMMC_ModelMain_V1.02_20200318.pdf
- DoD DevSecOps Reference Design - https://dodcio.defense.gov/Portals/0/Documents/DoD%20Enterprise%20DevSecOps%20Reference%20Design%20v1.0_Public%20Release.pdf?ver=2019-09-26-115824-583

Q&A / References

1. GitLab - 2021 Global DevSecOps Survey
   https://about.gitlab.com/developer-survey/
2. IBM – IaC
   https://www.thegreengrid.org/en/newsroom/blog/software-development-discipline-reshapes-infrastructure
3. RedHat - IaaS vs PaaS vs SaaS
   https://www.redhat.com/en/topics/cloud-computing/iaas-vs-paas-vs-saas
4. Forbes – Modern Shift Left Security
   https://www.forbes.com/sites/forbestechcouncil/2021/01/04/a-modern-shift-left-security-approach/