



Building a Security Program

(A Look in Four Dimensions)

Andy Ellis

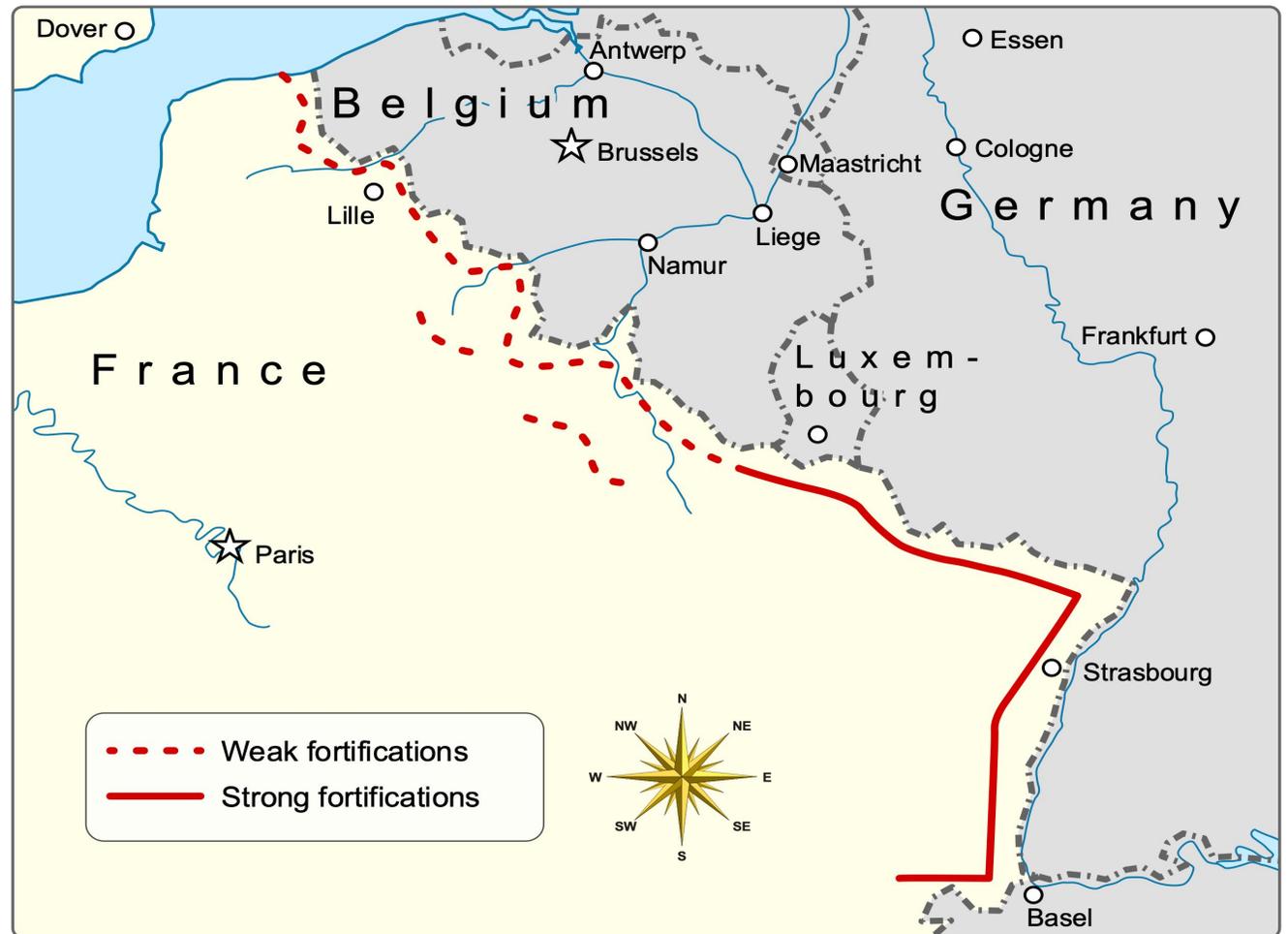
Advisory CISO, Orca Security





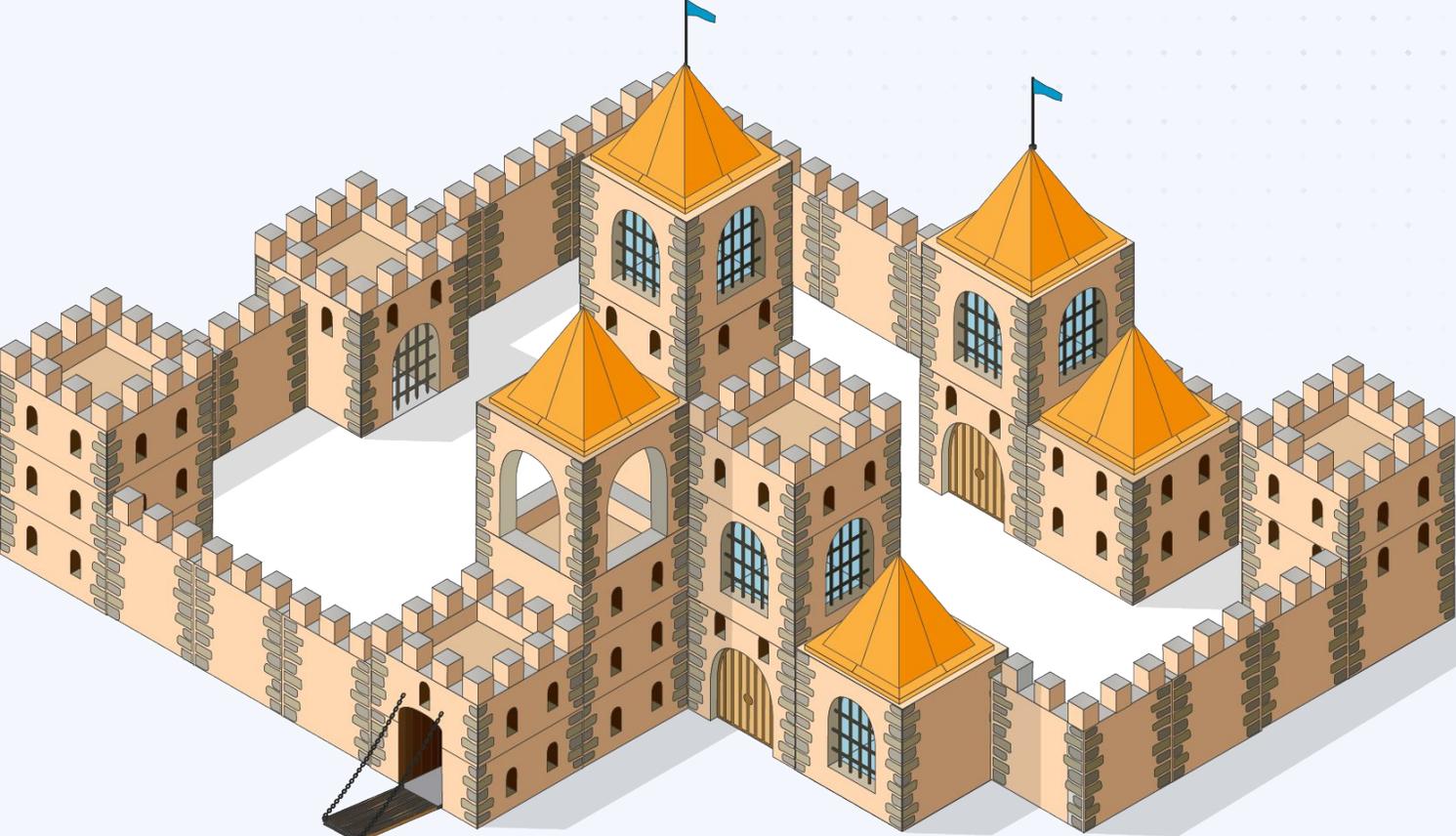
“Defense in Depth”

Maginot Line, CC BY-SA
4.0 Goran tek-en



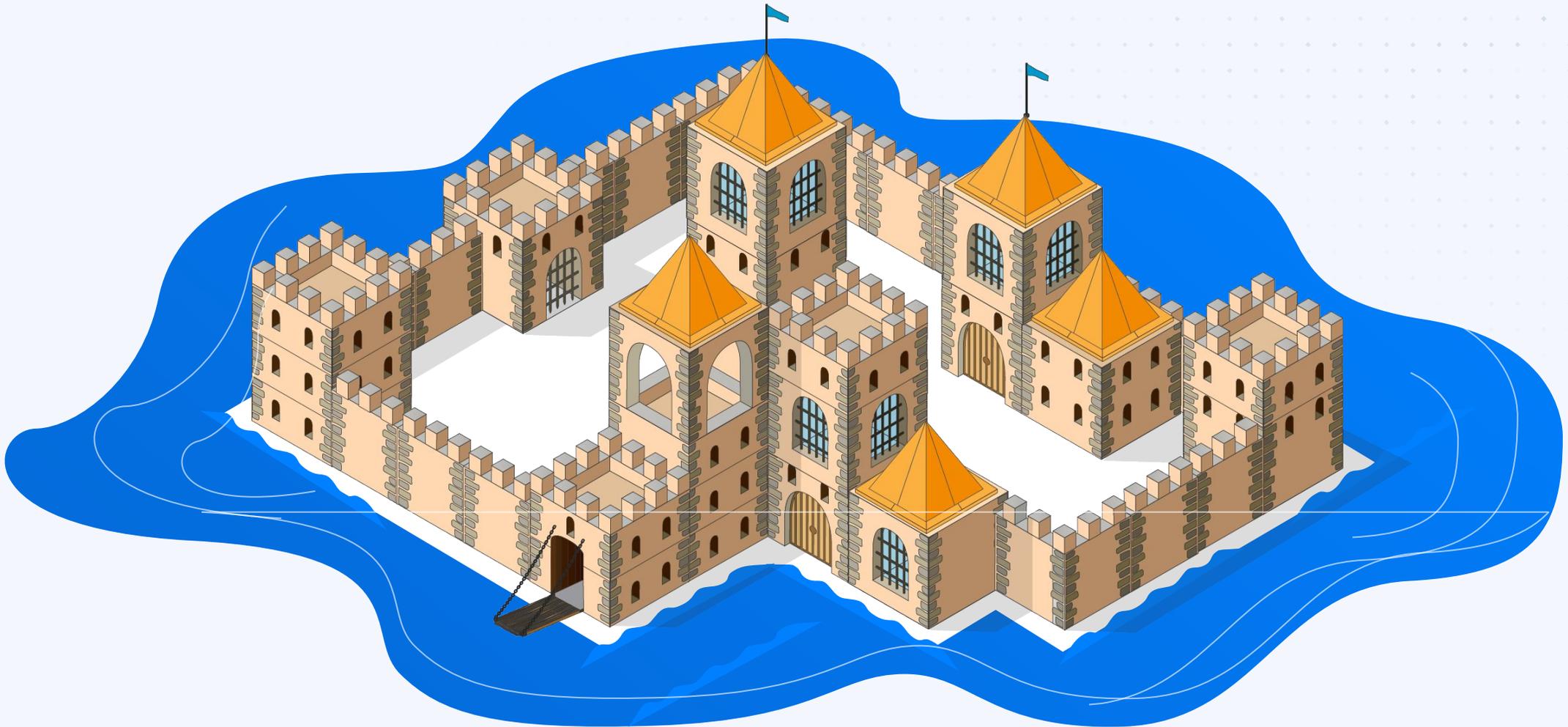


The Perimeter





The Moat





Defenders





Metric Exploration: Vulnerability Management

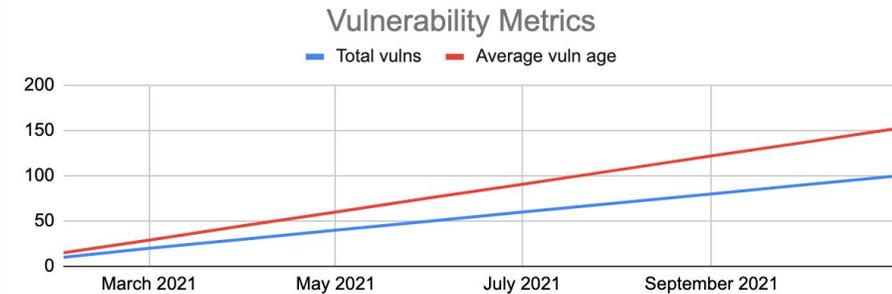
REVIEW THE CURRENT METRIC

Step 1: > Challenge the Definition

- > What systems aren't covered?
- > What vulnerabilities aren't counted?
- > What less relevant vulnerabilities are counted?

PATCHING VULNERABILITIES

> Average Age of Open Vulnerabilities



- > Definition: Defect measurement: How long have current vulnerabilities been unpatched?

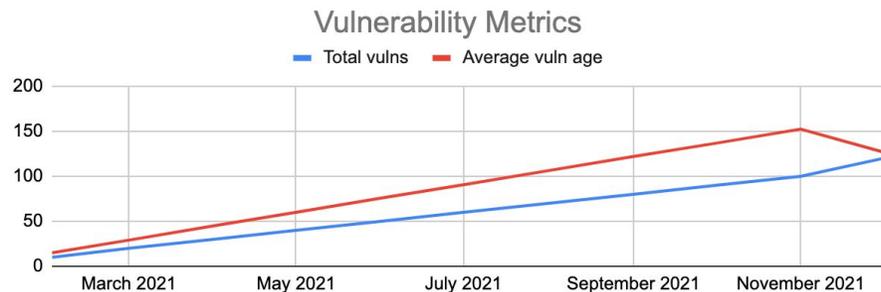


Metric Exploration: Vulnerability Management

BREAK THE CURRENT METRIC

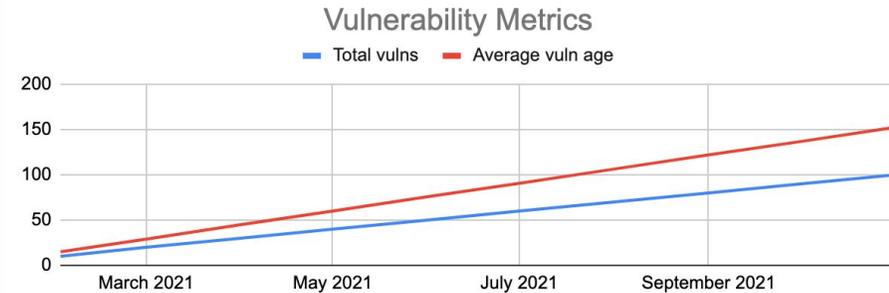
Step 1: > Challenge the Definition

Step 2: > Roundtable: What If?



PATCHING VULNERABILITIES

> Average Age of Open Vulnerabilities



> Definition: Defect measurement: How long have current vulnerabilities been unpatched

What if we don't patch at all?



Metric Exploration: Vulnerability Management

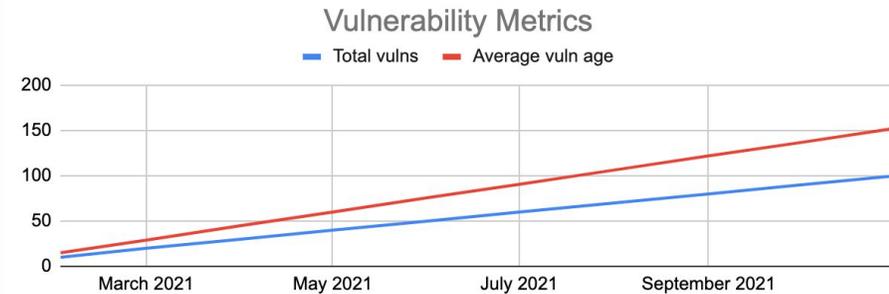
BREAK THE CURRENT METRIC

Step 1: > Challenge the Definition

Step 2: > Roundtable: What If?

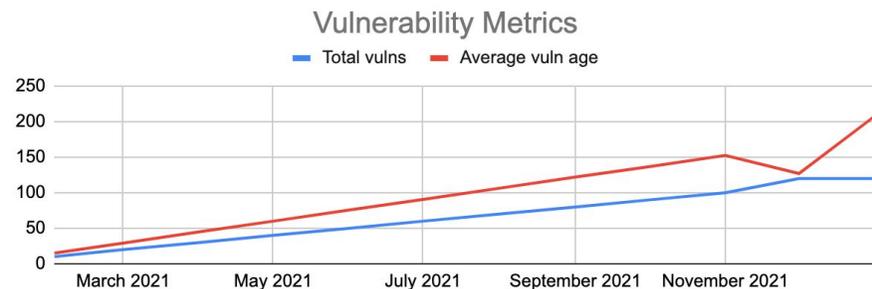
PATCHING VULNERABILITIES

> Average Age of Open Vulnerabilities



> Definition: Defect measurement: How long have current vulnerabilities been unpatched

What if we patch after a month?



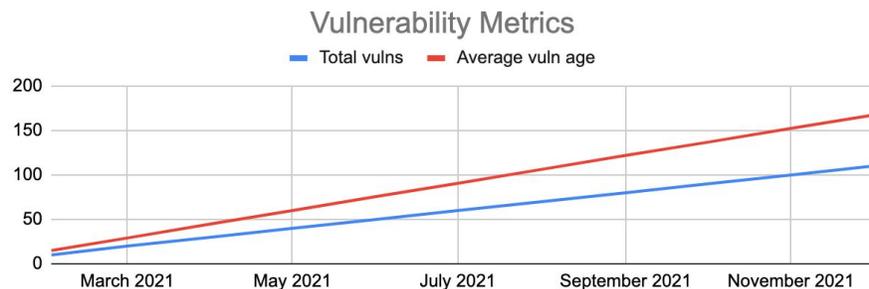


Metric Exploration: Vulnerability Management

BREAK THE CURRENT METRIC

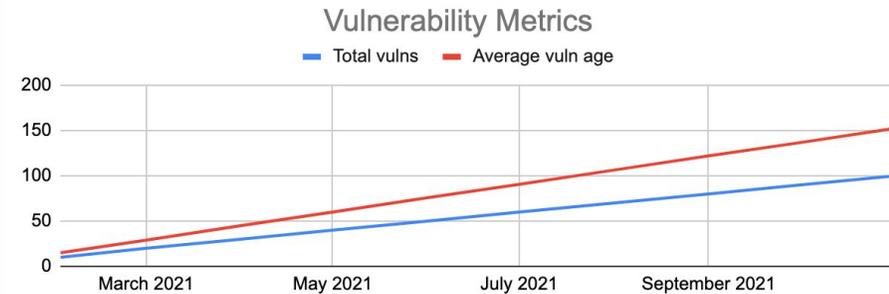
Step 1: > Challenge the Definition

Step 2: > Roundtable: What If?



PATCHING VULNERABILITIES

> Average Age of Open Vulnerabilities



> Definition: Defect measurement: How long have current vulnerabilities been unpatched

What if we patched between reporting windows?



Metric Exploration: Vulnerability Management

CONSIDER NEW METRIC

Step 1: > Challenge the Definition

Step 2: > Roundtable: What If?

Step 3: > Ask what you're trying to measure

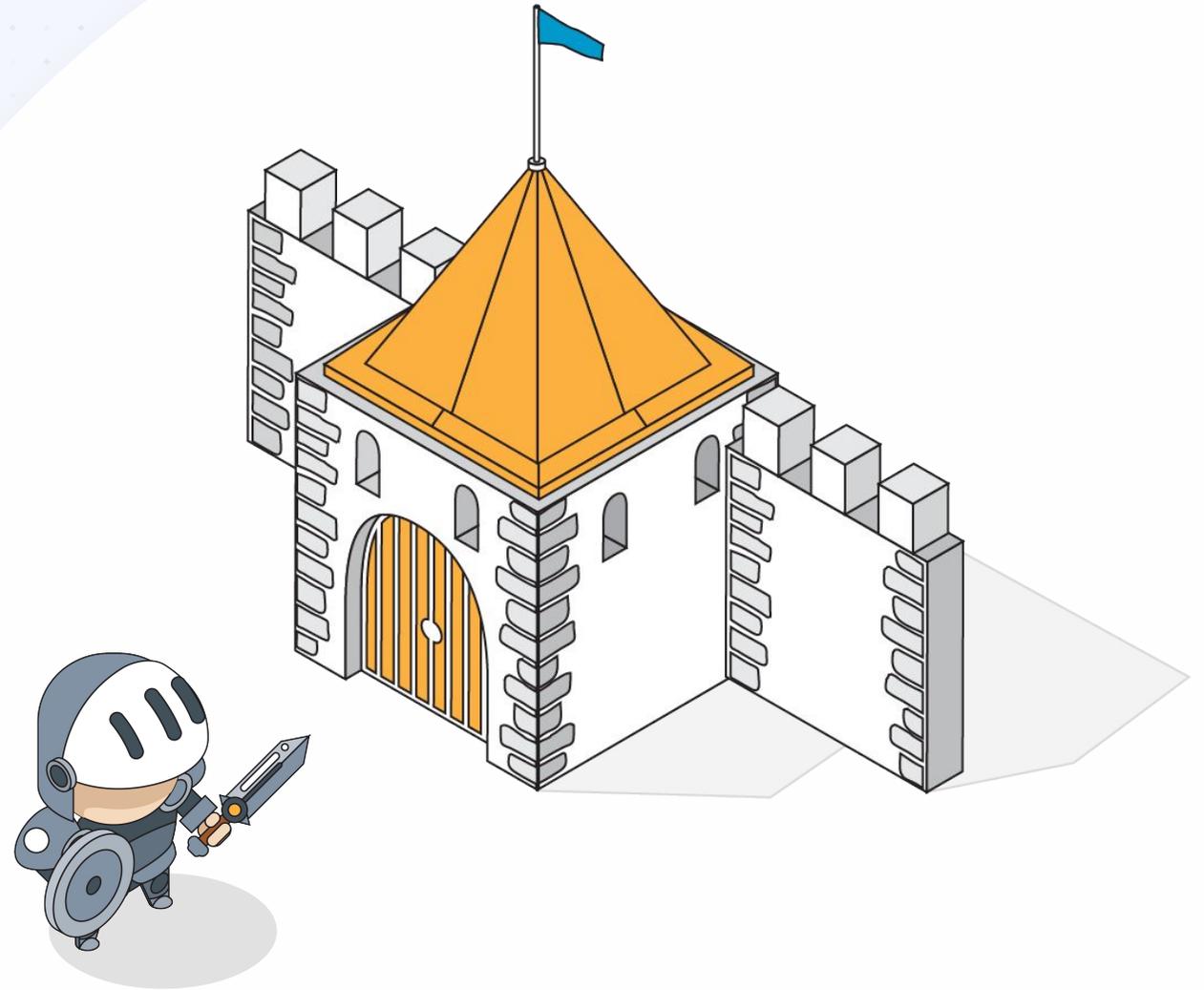
VULNERABILITIES

> Patch SLA measurement

Critical	High	Medium	Low
7 days	30 days	90 days	180 days
85%	70%	50%	40%

> Definition: How many vulnerabilities are patched within expected window?

Even in
“meatspace,”
defense
isn’t linear



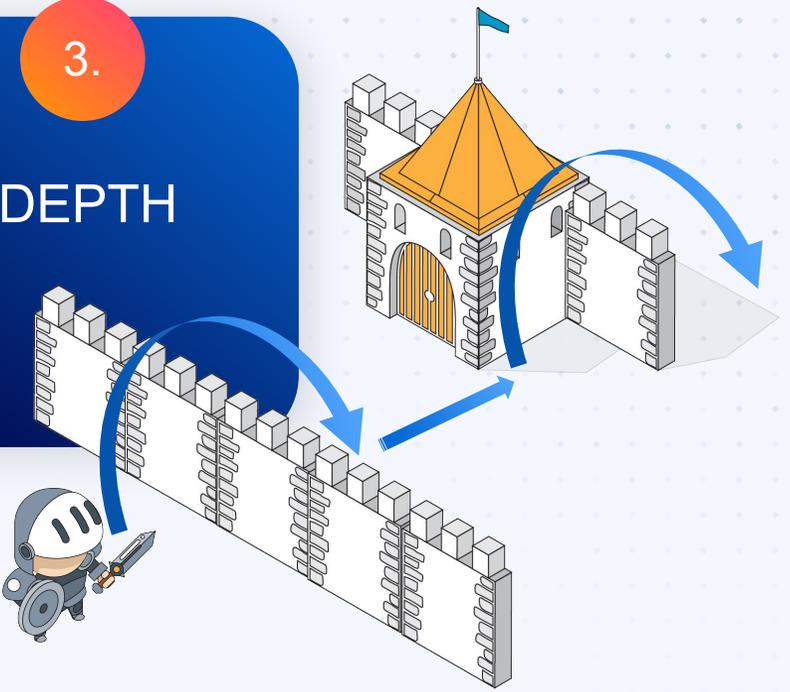
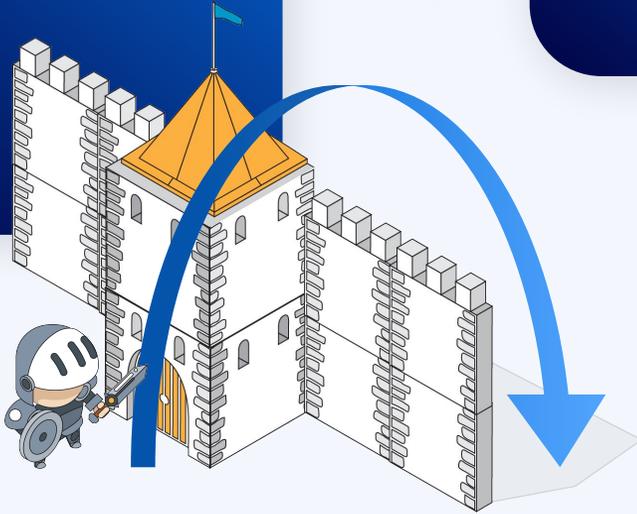
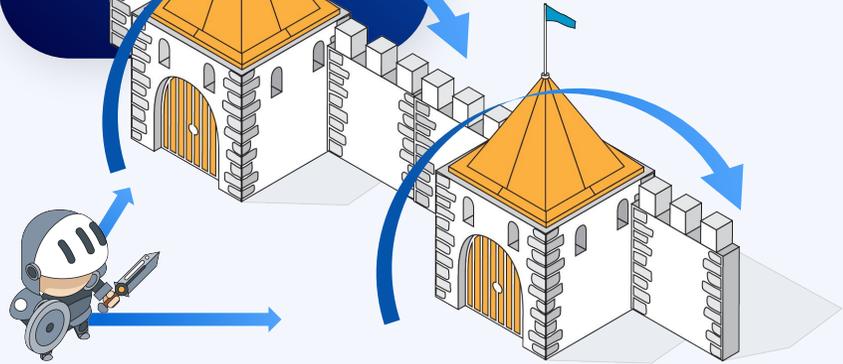


Attacks in ...

1.
BREADTH

2.
HEIGHT

3.
DEPTH





Defenses need to meet attackers...



Building a security program without considering how an adversary will try to penetrate it?



That's just a **Cyber Maginot Line.**

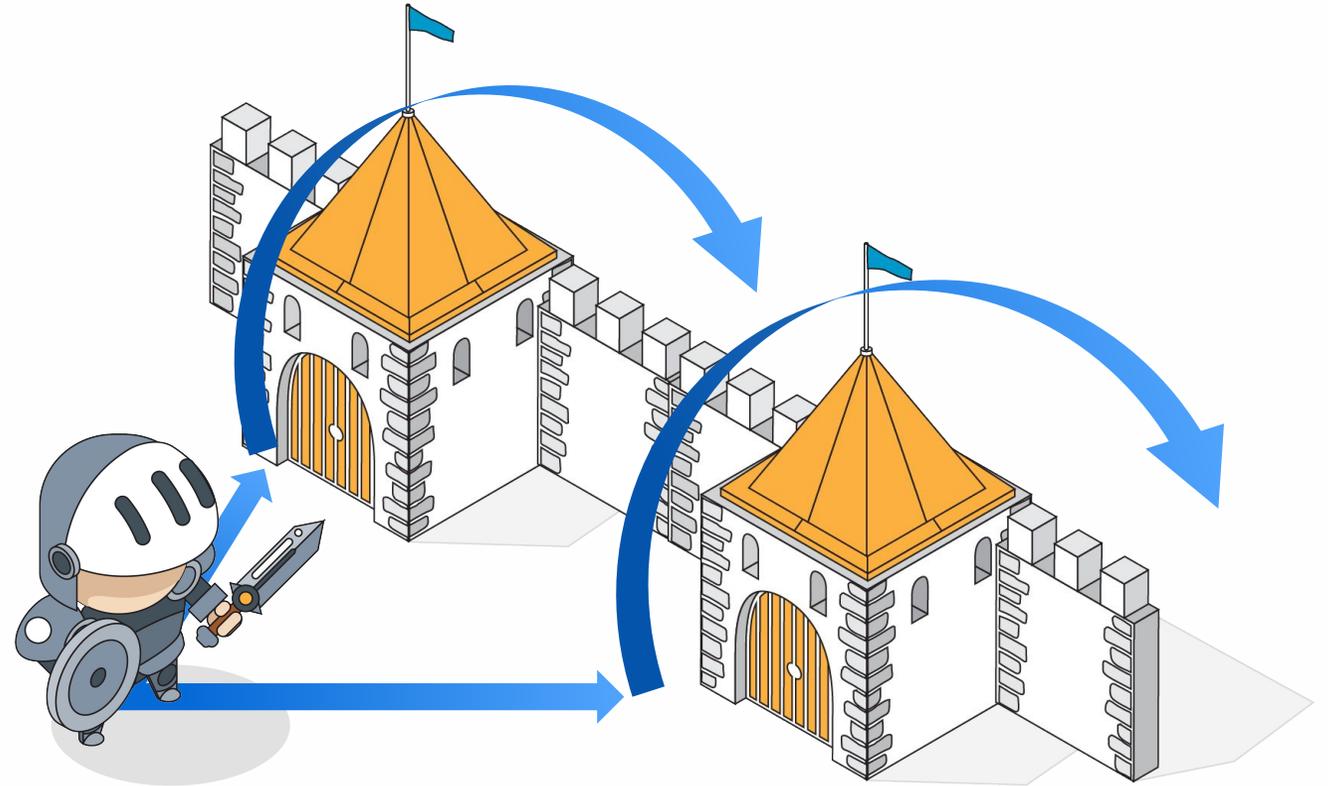


So how do we approach this challenge?

Dimension 1: Breadth / Width

Since the adversary can choose their point of entry:

- ✓ Defenders must have complete **coverage** of all of their assets, **especially** if they aren't well maintained.





Coverage: Asset Classes

Step 1:



List types of Assets

Step 2:



Count your Assets

Step 3:



Document ease of data collection

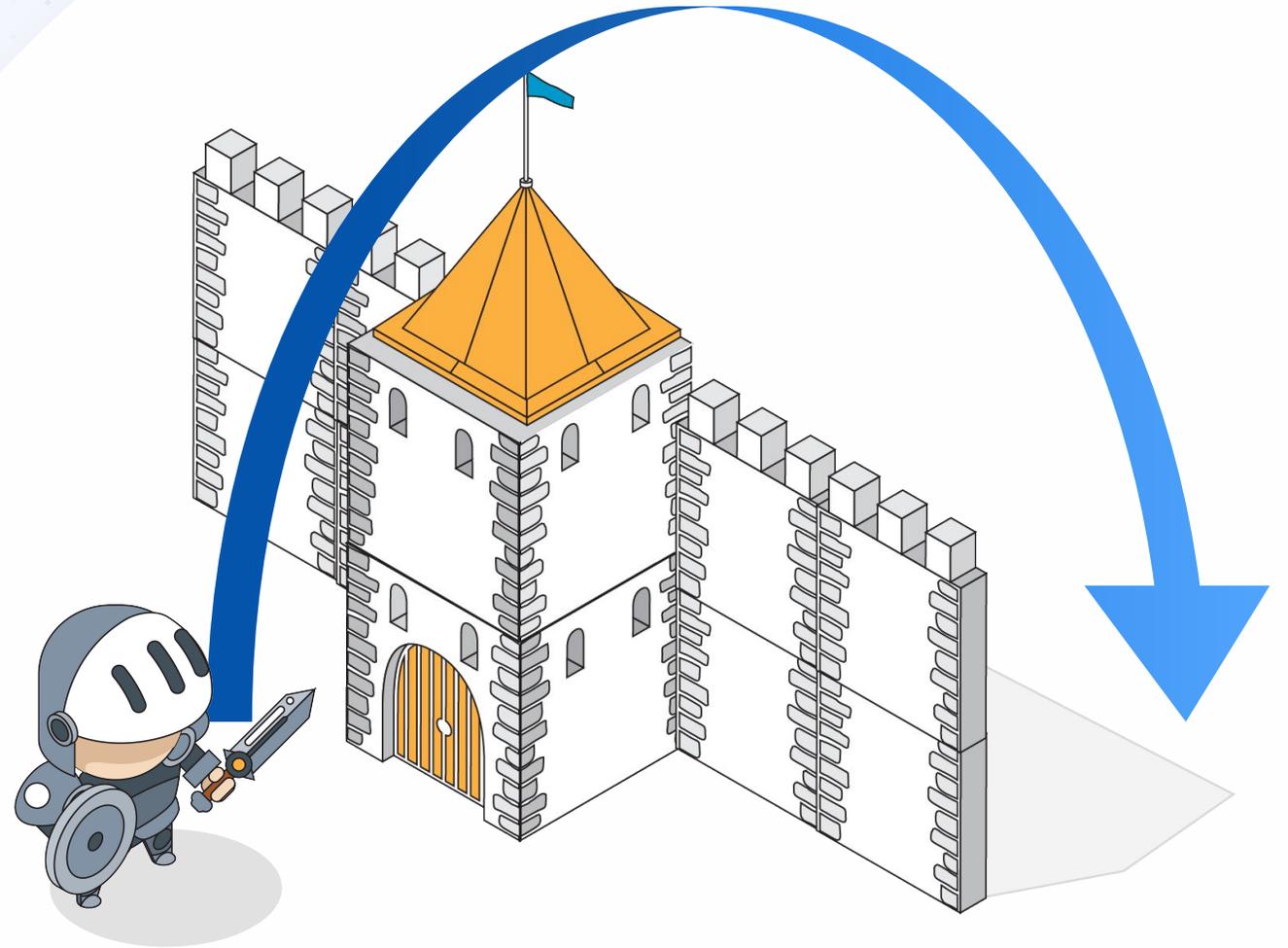
Public Cloud	152,435	🟡
Production Servers	3,000	🟡🟤
Dev/Build Servers	????	🟤
Enterprise Endpoints	9,267	🟡🟤
Enterprise Servers	352	🟡
SaaS Services	500+	🟤

- 🟡: Easy, automated
- 🟡🟤: Some manual effort
- 🟤: Lots of human effort

Dimension 2: Height

Since the adversary can quickly jump through security systems:

- ✓ Defenders must know how **comprehensive** their defenses are, and how they “stack.”





Comprehensive: Defenses

FOR EACH ASSET:

- Step 1:** Define Controls
- Step 2:** Define process measurements
- Step 3:** Document process maturity

PUBLIC CLOUD

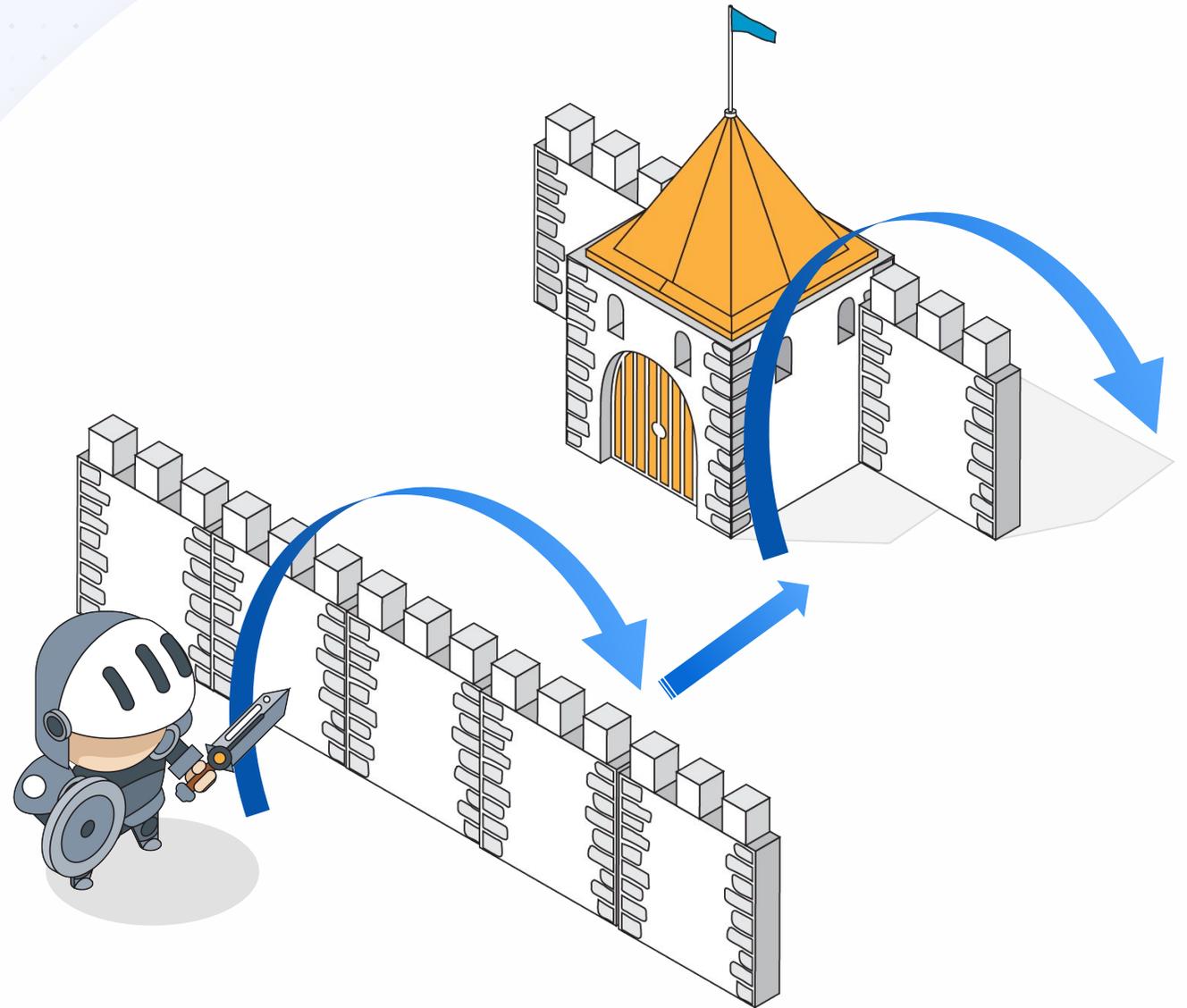
Inventory	152,435	🟡
Vulnerability Mgmt	@SLA 10% H/M/L: 7/30/90 days	⬛
Config Hygiene	High: 0 Med: 50 Low: 18,889	🟡
Authentication	User MFA: 100% Machine IDs: 50%	🟡
Access Control	Grants utilized: 82%	🟡
Exploit Monitoring	Dwell Time: 82 days	⬛
Data Protection	????	

- 🟡: No executive required
- 🟡: Some executive oversight
- ⬛: No process

Dimension 3: Depth

Since the adversary will laterally move in your environment:

- ✓ Defenders need the **context** of what is accessible to your front-end systems.





Context: Attack Scenarios

FOR ANY ATTACK TYPE:

Step 1:



Define effective defenses

Step 2:



Define incident response needs

Step 3:



Narrate existing controls in this context

RANSOMWARE

- > Stopped by:
 - MFA
 - Removal of lateral admin privileges
 - > Mitigated by:
 - Data backups
-

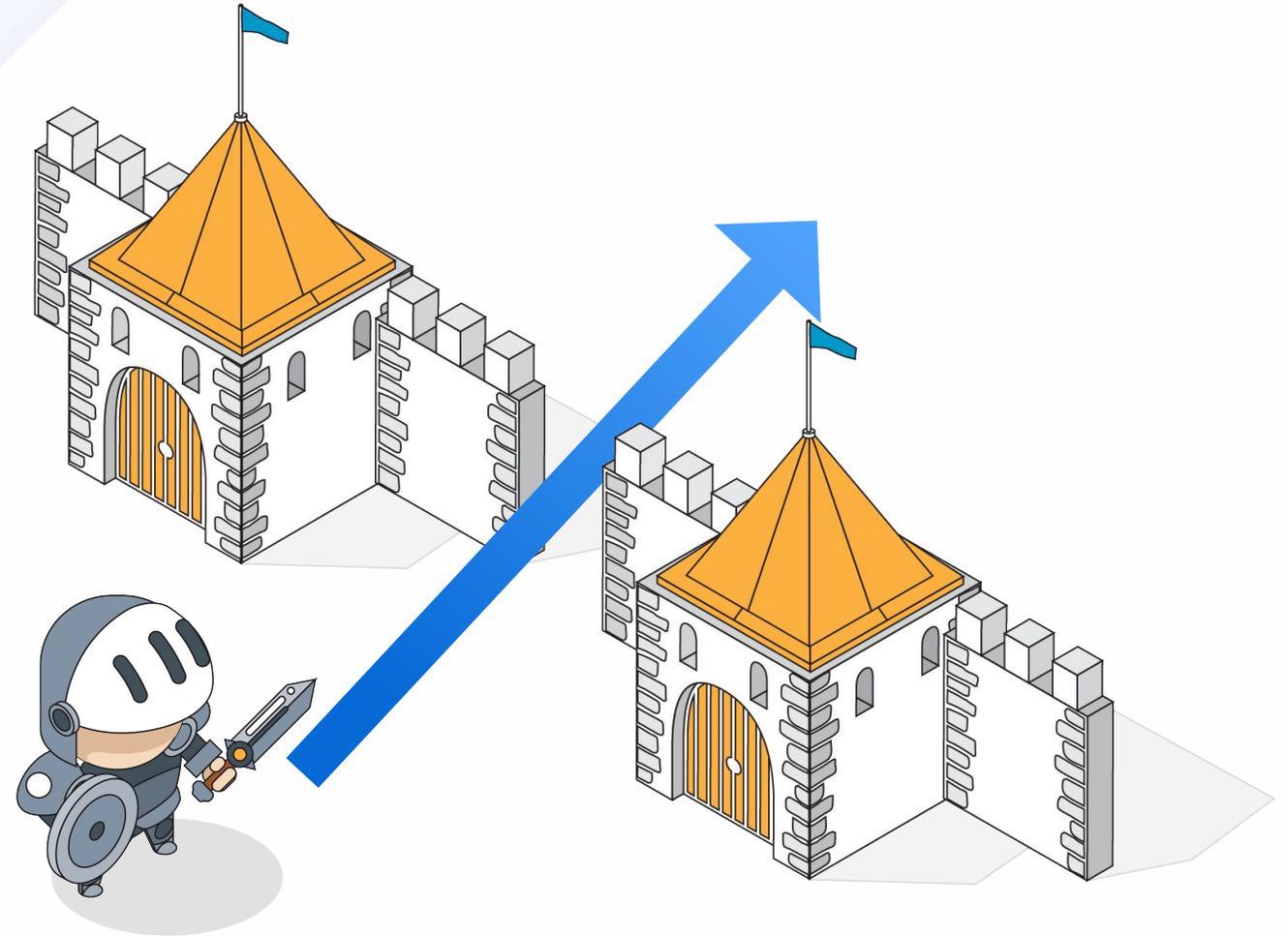
“We use FIDO-MFA, we’ve implemented three-tiered AD administration, and we’ve eliminated central jump servers.”



Dimension 4: Time

Since the adversary can wait until you aren't watching:

- ✓ Defenders need to ensure the **continuity** of all defensive controls.





Continuity: Do your processes mature?

FOR ANY SECURITY CONTROL:

Step 1:



Define and measure over-time efficacy

Step 2:



Define improvement “missions” to mature the controls

Step 3:



Track responsiveness to deviations from norms

VULNERABILITY

> Patch SLAs:

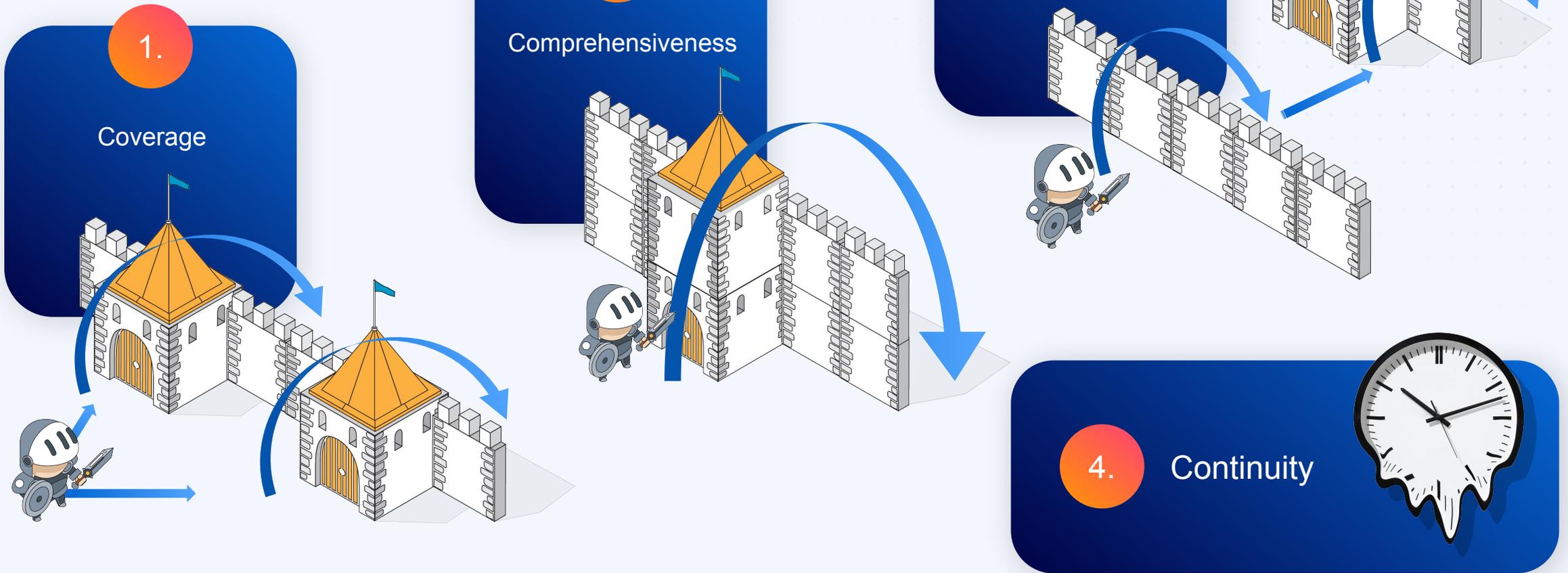
Critical	High	Medium	Low
7 days	30 days	90 days	180 days
85%	70%	50%	40%

> Mission: Improve build process to reduce software rollout latency by 5 days.

How many SLA violations were escalated before SLA was broken?



Defend with ...





Building a Security Program

(A Look in Four Dimensions)

Andy Ellis

Advisory CISO, Orca Security

