Appsec360

HOW TO

Build a product security program

Getting ready

Starting a product security program isn't just about running scanners or doing a once-a-year external penetration testing. Long before that happens, it's about enabling development teams to include security in their development workflow and be there to provide effective guidance when needed but knowing when to let them run the show. It's all about efficiency.

1

Know your apps, & gaps

You cannot secure the unknown. App & services Inventory is foundational to product security strategy. Start by building one. You may be surprised how many "cool" apps or services built during hackathons sneak into production.

2

Build a product security baseline

Build a picture of the current state of product security, how your company is solving the product security puzzle — and where the holes are — so you can start to formulate the best possible approach.

3

Optimize product security resources

Optimize the bandwidth of product security resources – it's key to a successful product security strategy. Build a decision engine that funnels releases into self-serve vs. fully served mode based on the security profile of the product or service.

4

Early visibility to security needs

Build a workflow that will provide development teams early visibility to security needs for their releases. Awareness removes friction & perceived stress of security.

5

Introduce the right security tools

Incorporate tooling that compliments the current processes that are comfortable for your development teams. What's essential is to inject security in ways that seamlessly fit into the dev workflows.

6

Develop cross-tool visibility

Introduce workflows that will reduce the signal-to-noise ratio of Appsec tools and automate workflows to establish cross-tool visibility.

7

Simplify Appsec issue remediation

Introduce workflows that will enable aggregation, deduplication, and compression of AppSec vulnerabilities to remove noise while streamlining findings for triage & prioritization based on business risk and impact.

8

Know your security debt

Not all Appsec issues will be remediated as they are found, some will be parked for resolution in a later release. Keep track of such exemptions so that they do not inadvertently expand the risk footprint.

9

Don't let developers be the weakest link

Keep a tab on the secure coding performance of development teams. Introduce training that equips your team to design & code securely all the way through your SSDLC so that you can reduce recurring vulnerabilities and ship quality code.

10

Continuous feedback loop across releases

Develop an effective, preferably automated mechanism to capture learning from past releases and enrich contextual risk awareness for subsequent releases. Reward good security work while enabling support for products that are not doing well in terms of security.