Your AppSec Program Training Course

From SheHacksPurple.dev

Assignment #1: Setting AppSec Goals
Your Name:

Below is a list of high-level goals for a generic application security program. These could apply to any office. But what about *your* office? What is important for your specific business? What issues do you know you are having problems with? Are certain types of incidents happening again and again? Think about this and write our two or three goals for your AppSec program

Here are a few potential goals you might set:
1) Teach developers about your top 3 security vulnerabilities in order to reduce their prevalence in your apps by 75%.
Possible supportive activities: start a lunch and learn program and have one event per month, for at least 3 months.

2) Stamp out XSS, completely.
Possible supportive activities: have a deep dive lunch and learn about it, send emails with info, add a test to every pipeline, create unit tests using the XSS Filter Evasion Cheat Sheet from OWASP

3) Improve the speed and way that Help Desk turns over security incidents.
Possible supportive activities: give the helpdesk team a short training on how spot an application security related incident, what to do and when they should call you.

4) Get a complete picture of your apps, and general idea of your app security posture.
Possible supportive activities: Do an application inventory exercise and do a DAST scan of 100% of them in the next 12 months.

5) Prevent secret spillage/ensure all secrets are managed as per industry best practices.
Possible supportive activities: implement secret scanning in every single pipeline, as of 6 months from now. Verify all secrets are in a secret store.

See how every one of those goals is extremely specific AND measurable? You will know in 6 months if you have given 3 lunch and learns or not. You might not be 100% sure you've eliminated all XSS in your org, but you can be certain you drastically reduced it and that you completed all of the steps to reach that goal.

Goals cannot be vague or unmeasurable. Here are some BAD examples of setting goals.
1) "Increase our app security by 20%" - That doesn't even mean anything. How can someone measure that? How would you do that? This can't be a goal.

2) “Get our developers to stop writing insecure code” – again, this is both vague and unmeasurable
3) “Ensure all of our software is secure” – No one can ensure that all software has zero vulnerabilities, and also, is that what they mean by ‘secure’?
4) “Have zero incidents related to software security” – this is a dangerous goal. This might make people not report incidents, for fear they will miss the goal. It’s also likely that if you think you have had no incidents that you are just not detecting them…. Which is worse than having to deal with them, as then you have no idea the damage that is happening…

Set your own goals for your team or organization. If you are currently out of work or a student, think of a previous place you worked or make up a theoretical workplace.

Note: you do not have to share these examples with the trainer or the rest of the students as it may contain sensitive information. If it contains sensitive info protect your answers appropriately.

Goal #1

Goal #2

Goal #3

- A complete picture of all of your apps; inventory. **Bonus**: alerting, monitoring and logging of those apps.
- Capability to find vulnerabilities in written code, running code, and 3rd party code. **Bonus**: the ability to quickly release fixes for said issues

- The knowledge to fix the vulnerabilities that you have found. **Bonus**: eliminating entire bug classes.
- Education and reference materials for developers about security. **Bonus**: an advocacy program, creating a security champion program, and repetitive re-enforcement of positive security culture.
- Providing developers security tools to help them do better. **Bonus**: writing your own tools or libraries.
- Having one or more security activities during each phase of your SDLC. **Bonus**: having security sprints or using the partnership model (assigned and/or embedding a security person to/within a project team).
- Implementing useful and effective application security tooling. **Bonus**: automating as much as possible to avoid errors and toil.
- Having a trained incident response team that understands AppSec. **Bonus**: implementing tools to prevent and/or detect application security incidents (can be homemade), providing job-specific security training to all of IT, including what to do during an incident.
- Continuously improve your program based on metrics, experimentation and feedback from any and all stakeholders. All feedback is important.