

TOURNAMENT FEEDBACK REQUEST

Audience: Developers



Greetings,

We've just completed our first Secure Code Warrior Tournament! Thank you to all those who were able to participate in this awesome secure coding event.

At this time, we'd like to invite you to fill out a short feedback survey about your Tournament experience.

WHY WE'RE DOING THIS

To provide the best training opportunities and experience for our teams, we need your feedback to hear what you think we did right and what we can improve on for future events.

This will let us to work together in shaping our secure code training program in the months to come.

WHERE TO FIND THE SURVEY

[ADD SURVEY LINK]

Thank you again for your participation and we look forward to hearing all about your experience.

If you have any questions in the meantime, please reach out to [NAME, EMAIL].

Good communication is central to a successful security training program. It can help generate interest from your developers while also keeping them in the loop with what's going on and what is expected from them.

The previous page is an example of what you can send out immediately after a Tournament has been completed. Immediately post-Tournament is the best time to gather feedback and suggestions as the experience is still fresh in everyone's minds.

For a list of example questions, [check out this article](#) from our customer knowledge base as a starting point.

Using a tool like Typeform or Google Forms can be a quick and easy way to collect, view and share the results.

This is a great way to let your developers know that you're listening to their suggestions about what they'll find most useful going forward.

If you're holding an on-site Tournament, it can be worth asking your participants to fill this out in the room together where it will also encourage some engaged discussion.

There's a lot of room to adjust information to make it relevant to your specific Tournament event and to your organization.

A few things to keep in mind when creating communications:

- **Be clear** - Provide clear expectations and directions
- **Be concise** - Avoid information overload or too many subjects in one message
- **Be informative** - Offer relevant and helpful information Devs need
- **Be transparent** - Be open with your teams about what's happening and why
- **Be empathetic** - Show that you understand your Dev's busy schedules

Keeping these things in mind will help you lockdown the key information behind the message you intend to share with your teams.