NULL SUJHAAV / PRASTAAV

Proposal for kick starting yet another of Null's formats - Null Sujhaav or Null Prastaav.[1]

Problem statement

From the security interviews that I have been involved with, as a candidate or an interviewer, for a few years now, for positions with product security teams, I feel that there's a gap between the demand & supply. While there's been a surge in the number of folks interested in getting in the security field & an equal thrust in the need & positive perception of security in product companies &/or the strart-up ecosystem, the available talent seems **slightly** misaligned.

Through this new format of null, we would like to propose an experiment towards solving the above problem.

Synopsis

The prod world, in my humble experience, is either not totally aware of the security nightmares, that the respective sec team is paranoid about, or for the lack of enough expertise/support/bandwidth, often choose to resort back to the short term, reactive solutions, like, just getting a pen test done on their product and then have their platform opened to an official bug bounty programme &/or have n number of automated solutions plugged in at multiple layers (with or without realizing that these solutions may not always fit the scene at all). Additionally, because of agile development, the rate at which products get released has been ever increasing. The problem is, the quick solutions, some of which have been sighted above, fall behind or insufficient in this race.

There has been a rise in the possible solutions to these problems, like dev sec ops being one of them, secure SDLC being another etc. (without getting into the discussion of which one is a subset of the other). The prod sec world, believes in leaning left; security by design principle as much as possible to support the ever fast growing needs of products, of course along with all the other solutions (sighted as quick above). Alone, these shift left approaches would also not be sufficient on their own.

Now, there's a huge pool of amazing bug hunters out there whose skills are supremely admirable. Similarly there's a genre of awesome pen testers, whose tech skills are simply unprecedented & daunting at times to be honest. However, the exact needs of product security and these skills are, at best, parallel lines at the moment.

[1] *'Sujhaav/Prastaav' means security recommendations, in this context*

The proposed solution

We would like to start another format of null, which we intend to call, *null sujhaav or null prastaav,* meaning recommendations (security recommendations to be specific)

Audience: All of the security community. Especially, folks interested in learning about product security
Modus operandi: semi promptu role plays
Participant 1: product developer or a product architect.
Participant 2: security engineer
Duration: Usually 30-40 mins per session

The general flow of the role play

1. We would be inviting developers/product architects to talk about a specific (hypothetical or otherwise) product that they have been working on or had worked on in the past or are planning to start work on.
2. They would need to present the architectural details of their design/product, the relevant components involved, flow diagrams, screen flows etc. whatever they can present. This need not be proper diagrams drawn with any UML tools etc. It could very well be on white board as well.
3. These architectures would need to be *preferably*[2] pre shared with participant 2 of the role play, the security engineer.
4. In front of the audience, during the role play, the developer/architect would give a *walkthrough* of the product/architecture to participant 2 (the security engineer)
5. The security engineer would try to get as best an understanding of the product as possible with a goal to secure the product from all perspectives. The security engineer may choose to ask questions as he/she feels during the course of this discussion.
6. After getting a proper understanding of the product, the security engineer would give tailored security recommendations, depending on the architecture & the related variables presented.
7. The developer/architect may choose to object/raise business/tech/priority concerns around the recommendations, in which case the security engineer would have to come up with alternative solution/s
8. This would be a free flowing dialogue/discussion without any pre-set flow of the discussion[3]

---

[2] *and hence it is a semi promptu role play.*
[3] *Why do we want to keep it that way? To give the audience a feel as close to real world as possible, because, IMHO, a lot of times, that's how the situations are, esp. in a startup environment. The audience should not feel like, "hey, he/she is able to do proper threat modelling around this because they have been working on it since ever". The fact is, despite*

How will it help bridge the gap

1. We believe that by doing many of these exercises, regularly improvising over time, we would be able to suggest another perspective, aligned with product security requirements, to the already existing pool of security talent.

2. This would also help the audience understand (& the community explore) the various possibilities in situations where the business needs to go live with the product despite the underlying risks.

3. This would, hopefully, also help with identification/generalization of security problems & possible brainstorming around more proactive solutions (as processes/tools/etc.) to these problems.

4. Assuming that we would succeed in getting developers/architects from various business/product/tech stack domains, this would help covering ground for many different types of use cases

Measuring the success/failure of the experiment

Some potential tools that we can use to validate our experiment & later measure it's impact:

1. Collection of feedback from the audience
2. If we are able to somehow collect data suggesting (probably with certainty) that people who have attended these sessions found it very useful to enable them to crack a product security interview, perhaps that can help.
3. We could start compiling all of these case studies, along with the security recommendations, into a gitbook (call it say, *The Big Book of Prod Security* or something) & start tracking stars on the repo. (or perhaps, even compile it as a hardbound copy & see if it sells enough copies ;))

Time frame for the experiment

We plan on doing this once a month, clubbed with null meets, to start with. If we see traction & enough demand we would try to increase the frequency. We plan to take at least 10 sessions to start looking at our metrics

---

that, the product designs, sometimes, are completely out of the blue and how to handle such situations best also needs to be showcased.