All the time, people talk about building a cyber security workforce and how it is a priority. I agree with the sentiment but disagree with most of the implementations because, despite all the money, energy, and love going into them, I see very little consideration being put into how effective the students of these programs will be once they graduate. **The goal of the Roppers curriculum is to produce more experts in the room, not more people who go to the experts for help.**

College degrees are the worst offenders because they intentionally do not develop expertise in exchange for a “well-rounded” education, but many bootcamps and similar programs focus their time on learning trivia and tools to pass certifications. Degrees and certifications lead to jobs (which is and should be the priority), but these graduates now have to fight to get themselves on the path to expertise, rather than being helped in the right direction during their education. With minor changes to curriculum design, existing programs can significantly improve the chances that the students they are putting out into the world become experts.

Running Roppers, a fairly large educational organization, I see a lot of beginners to the world of cyber. My real job is as a developer, and during part of that time I was in charge of my organization’s training pipeline, so I saw quite a few people starting their journeys. Spend enough time teaching people how Linux works, the basics of scripting, and the deeper, more complicated parts of computing, and you wind up seeing the same patterns and misunderstandings repeat themselves over and over.

Similarly, I am lucky enough to work with friends who are at the highest levels of the game which means I get to see technical excellence on a daily basis. None of us reached our current jobs via anything approaching the same pipeline, but the motivations and repeated actions that got us here are pretty much identical. To start with, we all wanted to be good at computers. Then we practiced with computers for a couple thousand hours until we were good at them. None of us felt forced into this field, we all love it and learned in our spare time.

There is not a security professional who is great at their job who didn’t start off their careers in love with learning new things and spending too much time learning about it. It’s not the most healthy, and people’s feelings on the subject change as they get more senior and more mature, but the experts are people who worked very hard because they loved the field. While there is an argument for training people who don’t love security to be competent employees, there is no argument that it is worth the time to attempt to train them to world-class expertise. Let students decide if they want to be great, and give them the tools to get there. If a student has no desire for expertise, they’ve made that decision for themselves, and no amount of education will overcome it.

Excellence in security requires understanding the technology. To secure something you must understand it, and be ready to defend it as the technology itself and the threats around you are constantly changing. On the other hand, to access something as an attacker you must often understand it better than the defenders.

To understand the technology is to understand how computers and the programs that run on them work. There is no way around this. You can't become a good hacker without understanding the technical fundamentals. Plenty of people and training programs try to skip the hard parts and jump straight to using tools they don't understand to accomplish tasks they don't appreciate. Then their scripts don't work and the kiddies are useless. There is no magic, just agreed-upon standards and deterministic code. Networking follows protocols, programs adhere to the application interfaces, and the kernel does what the kernel source says it is going to do. Nobody knows all those things, but they do know where to find why something is doing what it is doing (or have a friend they can ask).

To understand how things work is to embrace the complexity and be able to find where the answers are. Nobody knows everything. There is simply too much to learn for anyone to know everything they are going to face in the responsibilities of their job. Quick recall of best practices and where to find definitive answers is the key hallmark of an expert, not rote repetition of arcane commands. The expert knows the general search path to take and adapts their hypothesis as more information is learned or possibilities are eliminated. They rely on documentation and existing resources. If they don't know which resource contains the answers, they do the research required to find the resource. If they don't know enough to find a resource, they find another expert to point them in the right direction. If that resource doesn't exist, they create it and share it.

To find where the answers are is to not give up and to solve the problem efficiently. There is no book of knowledge, only hours, days, and years of experience looking for answers in a problem space and each time getting slightly faster at getting to the correct solution. The expert has seen harder problems before and has full faith in their ability to solve them, and has an idea of how long it will take (though this estimate is likely optimistic). The novice doesn't have the confidence required to spend the time on the problem it would take to find the solution, and even if they did, it would take them an excessive amount of time. Thousands of iterations of the problem-solving cycle is how the vast majority of the world's expertise was created. The primary difference between an expert and a novice is how long it takes them to find the correct answer and the consistency with which they get to that correct answer before giving up.

To solve the problem is to be good at hacking.
Hacking is about solving wide-ranging problems in wide-ranging technologies, learning on the fly when needed and creating a solution. ***Hackers don't view their skillset in terms of tools understood and programming languages they can write, but in problems they can solve.*** From the first moment someone starts in this field they have the potential to solve any problem, it's just a matter of time. There are short-term problems that an expert can solve every time and a novice will solve 0% of the time in the time required. However, given enough time, that novice can become an expert in that problem space and solve problems like that every time they come across them.

To increase expertise in novices, we can develop their ability to solve problems by giving them experience in the problem-solving cycle as early as possible in their journeys while managing

the complexity of the challenge. More information on this process can be found in this document How to Teach Towards Expertise in Security.

The skill of a hacker is basically correlated to how many iterations of this cycle they have done in the problem space they are working in, which basically just turns into a function of time spent hacking x problems to solve. This is what makes CTFs so effective at enabling the reps and sets required to be great because there are nearly infinite problems out there. Searching the internet and real life for real-world problems is significantly more work than hopping on your computer every weekend for competition and working through writeups during the week. While that might be a bit much for most people, by introducing CTFs and other quantifiable challenges that force problem-solving and allow students to compare themselves to their previous levels of ability, teaching organizations can minimally modify their curriculum to put their students on the path to expertise.