I Know Kung-Fu
An approach to technical self-assessment
By Keith Posner

If you haven't had a chance to read my recent collection of thoughts on hiring in information security, found here, think about taking a read. It is certainly not required reading but will hopefully touch on why I am so focused on technical skills. It will also provide context as I try and outline my approach to:

- understanding and baselining employee capability
- providing opportunities for continuous development and learning
- identifying and mitigating single points of failure across your team

Keep in mind there are always factors that can influence your ability to adopt certain practices. Aside from simple desire, two specific other considerations are:

- Your time – what this requires is upfront thought/planning relating to job roles and specific knowledge domains
- Your team's time – are you in a position to provide time/opportunities to staff for personal development including but not limited to independent research

In the end it comes down to your organization's view (along with your personal view) of the value of personal development, security and employee engagement. Now let's begin. Say hello to Max. You have managed to hire Max as an Intermediate Security Analyst. From the job description he is responsible for:

- Leading project risk assessments for web-based applications to ensure security requirements are in place, working collaboratively with architects, technical leads, and other stakeholders.
- Developing and delivering information security training to developers
- Miscellaneous stuff that is less important

Max's resume begins with a bulleted (and probably familiar) list such as:

• Web security • Industry frameworks • Application security testing • Risk assessment	• Vendor risk assessment • Strong development background • Excellent collaboration and communication skills

Having taken a few precious minutes to read my prior article on hiring in information security you have done your best validating that Max has a technical skillset. I use the phrase 'done your best' for a very specific reason. You can think of an interview as a short timeboxed discussion meant to judge suitability and fit from 5000ft. If you leverage any sort of technical skill evaluation as art of the interview process (pre or post), you will certainly improve your level of insight, but I will still argue you likely will be floating between 500-1000ft. Now the question is.... considering the high-level job description statements created for the role, what does Max need to know / be able to do to be successful.

Let's refer to this as applicable experience – and we will further link experience to capability. I call this crafting 'experience' domains.

I am going to suggest that you begin by outlining at a domain level what areas of experience are required for the role. Be specific and try to stick with meaningful terms. For Max, I am thinking the following might be appropriate given the job description...

- threat modelling / architecture risk analysis
- application security testing
- secure coding practices
- vulnerability management
- technology / security fundamentals

While Max may not be the one executing associated tasks within certain domains (e.g., penetration testing, secure code reviews), having the knowledge and experience will prove invaluable as Max participates in the associated processes (e.g., reviewing results, risk ratings, methodology, scope). Staying with the example above, if Max doesn't understand cross-site request forgery, and he sees a penetration report that indicates that it has been found and assigned a low risk, it may be tricky to understand if the risk rating is appropriate. We now have Experience domains. I expect that there will be between 8-12 of them. Instead of focusing on Max, why not define the knowledge domains across your entire team? Even if staff are not performing the same roles/ functions, you can map job roles to domains. It will come in handy, I promise.

Now let's focus some time on establishing capability levels and populating definitions. Ideally these definitions should be measurable but in certain cases it can be difficult. I am going to use an easy and familiar system of assigning competency. Feel free to come up with your own.

- Where's my belt
- White Belt
- Green Belt
- Black Belt

You could also use terms such as basic -> advanced, a numbering system, etc. Now, taking one of the experience domains we end up something that looks like a column within a larger matrix...

Capability	**Application Testing**
Where's my belt	• No experience
White Belt	• Familiarity and understanding of objectives, testing frameworks, common tools • Can document scope, methodology, results and recommendations • Has independently performed X # penetration tests on static website.

	• Has independently reviewed X # 3rd party prepared penetration testing report in the context of organization standards and best practices. Is able to identify issues with scope, methodology, results, ratings etc
Green Belt	• See above + • Add in more detail…more complex sites (e.g.,., logon)
Black Belt	• See above + • Add in more detail…more complex sites, mobile, full stack

So in this case, you can see the definitions are something that are measurable…."Has performed, Has reviewed". Is it easy to validate? Yes. For those definitions that are less measurable I would suggest a few approaches…

'Challenges' that validate experience / skillsets. They can be done in a fun and collaborative fashion or independently. For example, presenting a scenario to your team about a recent exploit (e.g., jquery file upload plugin). Ask them to walk through their thought process on identification, analysis, potential exposure and corrective action. If you want to make things even more interesting, get staff to bring their machines, take a few hours and have them work through the exercise and present their conclusions.

Another approach, assign strategic initiatives to team members who have indicated that they have certain experience or skillsets. For example, if someone has indicated that they have expertise in evaluating 3rd party security capabilities, ask them to create training content for junior staff including a side by side comparison of various industry assessment frameworks. If someone else has indicated they have coding / development experience, get them to introduce automation for manual tasks.

In the end, most things will end up being measurable. Also, you likely will have different expectations based on role / level across your team. For example, you will have different expectations for Max in terms of capability as Intermediate Analyst vs Ruby who is your Senior Analyst, even if they work in similar domains. As an example…

- Junior Analyst – White Belt
- Intermediate Analyst – Green Belt
- Senior Analyst – Black Belt

This establishes baseline capability expectations across your team and provides opportunity to align development plans to gaps in skill/knowledge capability. So keeping it simple, what does it all look like at this point:

Role Expectations:

- Junior Analyst - Grey
- Intermediate Analyst - Red
- Senior Analyst – Blue
- Security Technical Consultant – Yellow

*** Apply colors to boxes below or draw colorful arrows along the experience boxes

	No Belt	White Belt	Green Belt	Black Belt
threat modelling / architecture risk analysis				
application security testing	No experience	• Familiarity and understanding of objectives, testing frameworks, common tools • Can independently perform penetration test on static website. Can document scope, methodology, results and recommendations • Can interpret 3rd party prepared penetration testing report in the context of organization standards and	• See White Belt + • Add in more detail… • More complex sites (e.g.,., logon)	• See Green Belt + • Add in more detail… • More complex sites (e.g.,., logon) • API's

		best practices. Is able to identify issues with scope, methodology, results, ratings etc		
secure coding practices				
vulnerability management				
technology / security fundamentals				
Security & Dev Ops				
Automation & Development				

Have you ever thought, “wouldn’t it be great if I had something to guide new hires in training up in their new roles?” Well, I can tell you with the time spent establishing experience domains, capability level, definitions, role expectations, you are very close to having a training tool that is straightforward and measurable. Your team has obviously done work in the past. Hopefully, it was done well, and you can count on it for model answers. Map the work to applicable experience domains.

As an example, Frank, being one of your high performers completed a secure code review for an application (called AskME) 6 months ago. You have his analysis, code base, and recommendations. The application was of average complexity (let’s call it medium). On your training roadmap the secure code review of the AskME app would map to the Green Belt experience level. This means that anyone joining as an Intermediate Analyst will use this as part of their initial training efforts. The product of their work should come close to Frank’s model answer. Simple right?

So How all of this can be useful?
This is a self-assessment first and foremost, so it naturally ties into development plan discussions / goal setting. With self-assessment in hand, staff should be able to identify experience gaps, define action plans and show measurable progress in regular performance discussions. On the flip side, you will be able to ask about progress and the value of budget dollars spent. You can also use it as part of role progression discussions outlining baseline experience requirements as the basis for career path discussions. It can be the base of a new hire training program and provide opportunities for mentors and coaches on your team. With a training foundation in hand you can consider a talent pipeline creating defined paths of advancement.

Looking across your team, you will now have a view into domains where skills/experience might be lacking and/or where you might have an issue with one individual being a single point of failure. If your team is responsible for logging/monitoring and Frank is the only one who can understand security logs, and Frank leaves, you might run into a bit of trouble. The assessment allows you to direct work quickly and efficiently into capable hands. It can also help highlight where there may be lack of alignment between what staff believe to be their experience and their actual experience.

Final Thoughts
Keep in mind that if you decide to introduce some form of technical skill self-assessment, it is likely there will be more than a few gaps. Provide reasonable timelines for staff to bring themselves into alignment.
Keep an eye for staff who consistently undervalue or overstate their experience / capabilities and have open and honest discussions with them.

I am considering working on a technical self-assessment project where you would have the ability to select pre-existing domains (and create your own) with predefined skill levels and skill level definitions etc. If this is something that would be of interest, I would be interested in hearing from you discussion@halcyononline.ca.

..stay tuned