

Recipe for Culture of Security – 2 Cups Honey, 1 Cup Vinegar

Mary Ann Davidson, chief security officer, Oracle

One of my mother's favorite maxims is, "You catch more flies with honey than you do with vinegar." Getting people to support security is one of the biggest challenges a CSO faces, particularly because most of the people you need to believe in security do not actually work for you or in your organization. As you set out to create a culture of security, you must make it easy for people to "do the right thing." As a CSO, I've found that training and persuasion ("honey") usually work better than threats ("vinegar").

Often, the success of security programs is dependent upon someone other than the "high priests of security." Building a security culture is like evangelism: if you convert a sinner, he or she may tell the good news to other sinners, encouraging them to repent. Before long, the security heathen are converts and you can put both the honey and the vinegar back in the cabinet.

The Honey

Several years ago, external security researchers targeted one of Oracle's networking protocols. A developer in that team proudly reported that he fixed an externally reported buffer overflow (caused by ^A), only to be told by my team that he needed to make sure that he accounted for ^B, ^C, and so on. "But nobody would ever do that, would they?" he asked. When we assured him that the researcher would likely be back next week with all the different ^ variants he said, "But the code wasn't designed to handle that!" That, we told him, was the problem. After further discussions, including "see what I can do" demos by my internal hacking team, the developer was convinced that he needed to

assume a hostile – not benign – environment. Once converted, that developer became a source of light for his entire organization. Truth be told, he is sometimes now tougher on other developers than my assurance team is. As I look back, it proved to be well worth the time and effort we spent to convince him that these vulnerabilities were not theoretical and posed real consequences, especially because he created true believers out of his entire team.



The Vinegar

If the “honey” of education and patience can work wonders, the corollary to my mother’s folk wisdom is, “But sometimes you need to break out the fly swatter and mash the little &^%\$.” I frequently send my security bug wranglers to product development staff meetings to discuss open security bugs, hacking trends, year-to-date costs of fixing security bugs and to demonstrate the “hack of the week.” One week, a few developers who thought it was unreasonable to check related files for similar problems after they fixed a security bug heckled the team. From my perspective, this process is the same principle as checking for loose buttons on your coat as long as you are already sewing one back on. I felt duty



bound to ride into the next meeting on a broom and point out how much money one of our customers said they would need to spend if they have to patch every server (millions of dollars) and furthermore, that when I sent my team in to share their expertise, I expect them to be treated with respect. <Attitude> problem solved.

On another occasion, we had an entire product team look for the “top four” most common security bugs, based on short handouts developed by the security team (what they are, how to prevent them). Every development group (but one) found issues, and the combined group fixed a total of 80 bugs before shipping the product. Later, my internal hacking team found a number of serious issues in the code of the team that “hadn’t found anything” because the development manager simply hadn’t looked. As result, the manager did not receive a raise, bonus, etc., in the next compensation round. The message was: “We made it easy for you to do the right thing in security, and because you didn’t, you will be held accountable.”

One of the ways you can help yourself check for the vermin level is via judicious use of security metrics. I am reminded of a

true story from the annals of World War II, when some bored sailor at Naval Station, Somewhere-in-the-South-Pacific got tired of all the paper work he filled out daily for his command headquarters. He made up a new form, and started reporting the success rate of the flypapers in the mess hall: flypaper B1 trapped and retained 13 flies, flypaper A7 did even better: trapping and retaining 18 flies. Before he knew it, headquarters was demanding “flypaper utilization reports” from everyone in the command. (This is how bureaucracies flourish.) All joking aside, you can’t improve something you can’t measure, so actually measuring things like mean-time-to-fix bugs helps you identify problem areas. You do need to ensure you are measuring and rewarding the right things, however, otherwise people tend to “manage to metrics” and you could inadvertently be rewarding the wrong things.

For example, while in general closing bugs in a hurry is an admirable quality, the relevant metric for security bugs isn’t merely “how fast did you close this?” but “how thoroughly did you address the issue?” In that sense, what you want is for security bugs to never be “re-opened,” that is, investigated and fixed again because it wasn’t done the right way, the first time.

If the “honey” of education and patience can work wonders, the corollary to my mother’s folk wisdom is, “But sometimes you need to break out the fly swatter and mash the little &^%\$.”

In short, the purpose of metrics isn’t just finding the correct fly to swat, it’s trying to identify where problem areas are so you can figure out why the problem is occurring, and focus your limited resource on solving the worst problems first. For example, if a development group’s “mean time to fix security bugs” is lengthening, it could be because a single developer is sitting on a bunch of critical security bugs. Or, it could be that several critical people quit and the team is under resourced. Numbers tell you part of the story, and can be used to punish the guilty, praise the virtuous, or in some cases highlight a trend that nobody knew was there. Given that the nature of security people is always to say “we aren’t doing enough,” metrics can help you figure out “how good does good have to be,” and target your efforts when “good isn’t good enough.” It also helps you measure how much honey and vinegar to use, and who gets which condiment.

A last critical point to the use of honey and vinegar is that your security program will fail if there are no consequences for bad behavior; for example, if you write security policies that aren’t followed, and there is no enforcement, you might as well save your efforts and not write them in the first place. In my experience, most people will do the right thing in security if you teach them what the “right thing” is, make it easy for them to do, reward good behavior and punish bad behavior. Honey works best nine out of ten times but sometimes you need a judicious application of a fly swatter to keep the vermin level down.