Katie Moussouris is the chief policy officer for HackerOne, a San Francisco–based startup that developed a vulnerability management and bug bounty platform. Moussouris oversees the company’s philosophy and approach to vulnerability disclosure. For seven years before that, she worked at Microsoft, ending up as a senior security strategist. She also headed the company’s bug bounty program, served as content chair for its BlueHat internal hacker conference, and helped with its secure development life cycle process.

You started on a Commodore 64 programming in BASIC. What’s the coolest program you wrote as a kid? It was probably my first one. I was really into Choose Your Own Adventure books, and I wrote a text-based adventure game. That was really fun for me.

Did you use and adapt some code from other people? I think I took some examples from a book on case statements. I definitely borrowed some code. I’m sure it had bugs in it.

Then you decided to study biology and made a quick switch back to the computer field. I wanted to cure cancer and AIDS. I got to work on the Human Genome Project, which was great. But I found myself going back to the computing world. I made a couple different career shifts, working as a systems administrator for a while. I learned the pain of having to defend your own network from attackers and polished my penetration-testing skills doing that when I worked at MIT.

I became a Linux developer when I first moved to California. I was mostly working with the different systems administration tools, fixing a lot of bugs. I realized we didn’t have a formal computer security response program there, so I started one. That’s where I began my career in security response and vulnerability handling and coordination.

Tell us in very basic terms how a bug bounty program works. A bug bounty program is a cash reward offered by a vendor in exchange for vulnerability information. It’s typically offered on a per-bug basis: one bug, one bounty. But a lot of the incentive programs I worked on, for example, at Microsoft, were looking at broader things like learning defensive techniques and new attack or mitigation bypass techniques. That’s essentially a new way to exploit and bypass all the safety measures on a platform’s latest version. These were more structured incentive programs that looked for a higher-order outcome than you look for with basic bug bounties.

Is this what you do at HackerOne, or is it more bug bounty related? HackerOne is a platform and toolset that enables response teams to receive vulnerability reports. You can do that on our platform with or without a bug bounty. We’re really about vulnerability coordination. The platform is free if you want to use it for vulnerability coordination. We charge a fee only if there’s a bounty involved. We also advise our
About Katie Moussouris

Katie Moussouris is the chief policy officer for HackerOne, a platform provider for coordinated vulnerability response and structured bounty programs. She oversees the company’s philosophy on vulnerability disclosure, advises customers and researchers, and works to legitimize and promote security research to help make the internet safer for everyone. After working at MIT, Moussouris joined Symantec and later Microsoft, where her work encompassed industry-leading initiatives such as Microsoft’s bounty programs and Microsoft Vulnerability Research. She is also a subject matter expert for the US National Body of the International Organization for Standardization (ISO) on vulnerability disclosure and handling processes, secure development, and penetration testing.

customers on the best way to create an incentive program if that’s what they’re ready for. You really have to do a lot more security homework for a bounty.

Bug bounties are something we saw emerge in the BSIMM [Building Security In Maturity Model] community (http://bsimm.com). It makes sense to have a bug bounty or vulnerability coordination system as a software vendor, but it’s iffier when it comes to financial services and healthcare firms. What do you recommend for those sorts of firms? Financial services and healthcare firms—really any organizations or verticals at high risk of cyberattack—already hire penetration-testing companies. So they’re very comfortable with the BSIMM, including the idea of bug bounties. Instead of looking at it as, “I’m only comfortable hearing about vulnerabilities from a specific small set of individuals that I hire,” we want to open up that mindset to, “If anybody could tell me about a vulnerability in my product, I’m interested in hearing about it. And I’d rather hear about it first than wait for an attack to happen.”

I was going to ask whether it’s better to use a HackerOne system or hire an internal penetration test team. But I know you’re going to say both.

It’s absolutely both. There’s a time and a place to do different kinds of testing as you’re developing the software. You and I are both big advocates of secure software development, and there are appropriate times and methodologies and tools to use when you’re testing along the entire life cycle. Using external folks of any kind, whether they’re penetration testers under contract or the unwashed masses of the Internet, you really just want to know if there’s any way to break into your system.

Do some of the researchers or hackers that you coordinate with work on all 68 bug bounty programs? Do you have a stable of people who do that? Our platform has a built-in reputation system for hackers. And we’re very transparent about how reputation is affected by different outcomes of a hacker reporting a vulnerability. You can imagine that their reputation improves when they find an issue that’s serious enough to award a bounty. But to answer to your question, there are folks who are very prolific. They came in looking at one piece of software—maybe they came from Russia because they were looking at a specific piece of Russian software that was hosted on our platform for vulnerability reporting. And then they ended up spreading their knowledge and finding a lot of vulnerabilities in many other people’s programs.

Let’s talk about breaking and building in security. Many conferences focus on breaking things and getting bounties. Is it possible to turn the people who do the breaking into builders who can develop secure systems? Do we even want to do that? I think it’s possible. As much as we talk about getting people to think more like attackers, getting attackers to think like defenders is as hard or harder in a lot of ways. Generally speaking, they don’t have to worry about the piece of software functioning in exactly the way the user or creator envisioned. But that functionality is going to be a huge part of securing software—you have to secure it so that it still works.

An example of incentivizing this switch is the bounties that I set up at Microsoft. There was a mitigation bypass bounty and a defense bounty. If you came up with a way to defeat all the mitigations but also came up with a way to defend against your new attack technique, you got more money.

This actually was paid out for the first time to researchers who work at ZDI [Zero Day Initiative]. Three bounties were launched in 2013, including one defensive bounty. Just a few months ago, ZDI’s security researchers not only got [US]$100,000 from Microsoft for a new mitigation bypass but also claimed the very first defensive award of an additional $25,000. I thought it was cool that somebody actually created that shift in thinking from being a breaker to thinking, “Here are some viable ways that the platform could introduce measures to defend against this.”

So, Microsoft gets something for $25,000 that might cost a million bucks in the open market, if it were a real technology transfer.
Who else except for Microsoft would buy a defensive technique? Looking for those gaps in markets and market motivations was definitely one of the things that went into my thinking when creating Microsoft’s bounties. Create incentives for what you want and pay attention to those gaps in the market where you can develop an incentive for something that’s useful only to you.

I’ve always wondered whether you should teach developers to think like a bad guy. I used to think that you should, but then Microsoft’s Steven Lipner changed my mind and convinced me that you can’t teach all developers how to think like a bad guy. What do you think?

You’re right—you can’t teach them all. I think one of the biggest problems I had as a penetration tester was convincing a developer that somebody would think like a bad guy.

They look at you like you killed their puppy.

Exactly. Who would kill my puppy? It was made of software. It was beautiful. So, convincing them of the likelihood that somebody would [think like a bad guy] is often the biggest challenge.

What’s the most effective way you found to do that?

Unfortunately, you actually do have to show them something exploitable most of the time. I can’t tell you how many times I’ve actually had to deliver an exploit to somebody when we were trying to tell them about a vulnerability. Then you have the danger of them thinking that exploit is the only vector.

Tell me about some of your mentors. What did they help you with?

My mentors have really run the gamut. Bob Bruen was my boss for a brief period and my friend for a lot longer when I worked at MIT. He would tell me if some crazy idea I had was actually viable and give me different strategies for making it happen. When I was working at MIT on network planning, I basically outlined how we should do it, but the folks were kind of politely and not so politely dismissing my advice. So I went into Bob’s office and said, “I need to borrow your gray beard for a minute. Will you just tell them this, that, and the other thing?” And he said, “Look, I can tell them, and you’re right that they’ll probably listen to me. But eventually, you’re going to have to grow your own gray beard.”

He basically said that at some point, you’re going to have to be able to communicate and command authority in areas where you have expertise. It’s not always going to be easy, and you’re right that I’m going to have an easier time walking into a room and having my expertise assumed, whereas you’re going to have to work a little harder to establish it. But you can do it. I believe in you. And that was that.

Ever since then, I’ve been growing this gray beard. I’m kidding. But I’ve tried to make sure that I lead with my knowledge and try to be helpful in a situation first. And usually that ends up working. If you look young, you’re going to have to lead with your authoritative foot.

I’ve had a lot of experiences with peer mentors. It ends up that your whole network is a mentor, in a way. You decide where you want to steer your ship, and then you go to your network of peers or colleagues and say, “I’m interested in moving into this area. Can you make some introductions?”

This past year at HackerOne, I’ve been fortunate in being able to take my expertise from vulnerability disclosure and coordination and parlay that into speaking to more and more policymakers and lawmakers about things that might affect security research, like the recent proposals to expand the US Computer Fraud and Abuse Act. This year has been about me leveraging my peer mentor network and getting introduced to people who are willing to hear about what this might mean to security.

One of the biggest problems I had as a penetration tester was convincing a developer that somebody would think like a bad guy.